

Buku Anilisi Numerik

by Davi Apriandi

Submission date: 21-Jan-2019 06:03PM (UTC-0800)

Submission ID: 1066811515

File name: Analisis_Numerik_Simulasi_MATLAB_E-Learning.docx (3.23M)

Word count: 19027

Character count: 134353

LINGKUP PENDIDIKAN MATEMATIKA

ANALISIS NUMERIK

Pemograman MATLAB berbasis Simulasi E-Learning

Penulis 1 : Reza Kusuma Setyansah., S.Pd., M.Pd.

Penulis 2 : Davi Apriandi, S.Pd.Si., M.Pd.

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 kehadiran 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 diletakan 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 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 Pemograman 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.

Kata Pengantar	i
Daftar Isi	iii
Daftar Program Simulasi.....	iv
Mekanisme Simulasi E-Learning.....	v
SIMULASI 1 : PENDAHULUAN	1
SIMULASI 2 : GALAT	29
SIMULASI 3 : AKAR PERSAMAAN TAK LINIER	52
SIMULASI 4 : SISTEM PERSAMAAN LINIER.....	90
SIMULASI 5 : ANALISIS REGRESI DAN INTERPOLASI	131
SIMULASI 6 : INTEGRASI NUMERIK.....	175
Daftar Pustaka.....	194
Tentang Penulis.....	195

Daftar Program Simulasi

- ✓ 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

Mekanisme Simulasi E-Learning

Perhatikan langkah dari mekanisme penggunaan untuk memperoleh aplikasi simulasi e-learning 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 masukan dari ribuan pemakai melalui laman <https://www.mathworks.com>. Dalam lingkungan dunia pendidikan ilmiah menjadi alat pemograman 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 metoda numerik.

Salah satu aspek yang sangat berguna dari MATLAB ialah kemampuannya untuk menggambarkan berbagai jenis grafik, sehingga kita bisa memvisualisasikan 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 pemograman 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 aritmatika. 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 pemograman 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, Differensial 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) ²²Lingkungan Kerja MATLAB

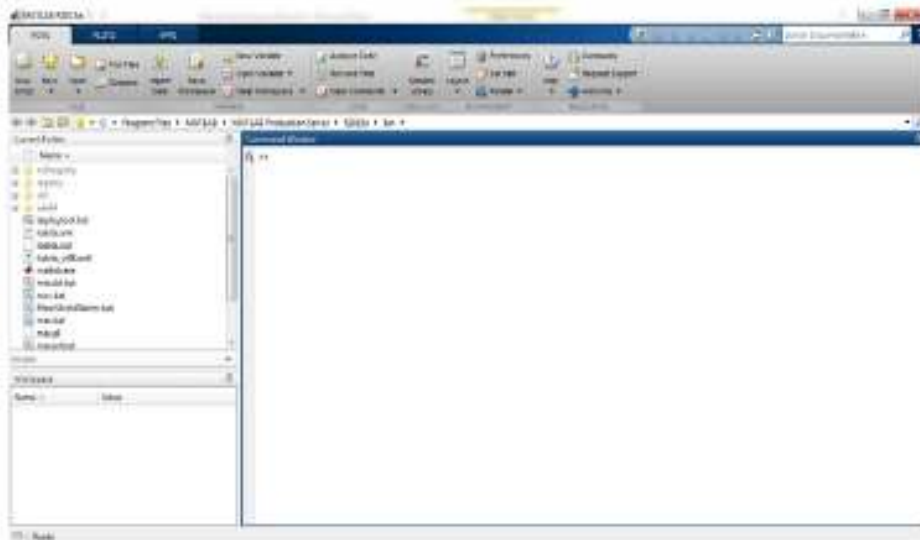
Dalam bahasa pemograman lainnya, MATLAB juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam pembangunan aplikasi. Pada setiap update ¹⁷versi terbaru MATLAB, bentuk lingkungan yang dimiliki semakin dilengkapi. Lingkungan terpadu ini terdiri dari *form/window* yang memiliki fungsi masing-masing. 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 pemograman MATLAB dalam buku ini

menggunakan versi R2015a atau disebutkan dalam versi terdahulunya dengan versi 8.5.

17

Window Utama MATLAB



Gambar. 1.1. Tampilan Window Utama MATLAB.

Window ini adalah window induk dari pemograman 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.

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.

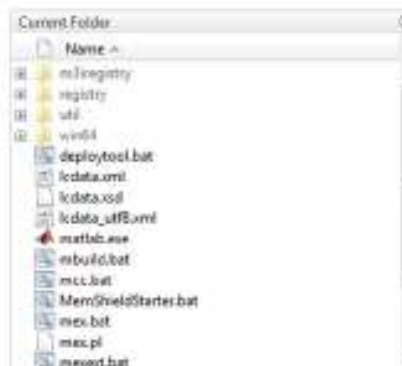
17 Workspace Window



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



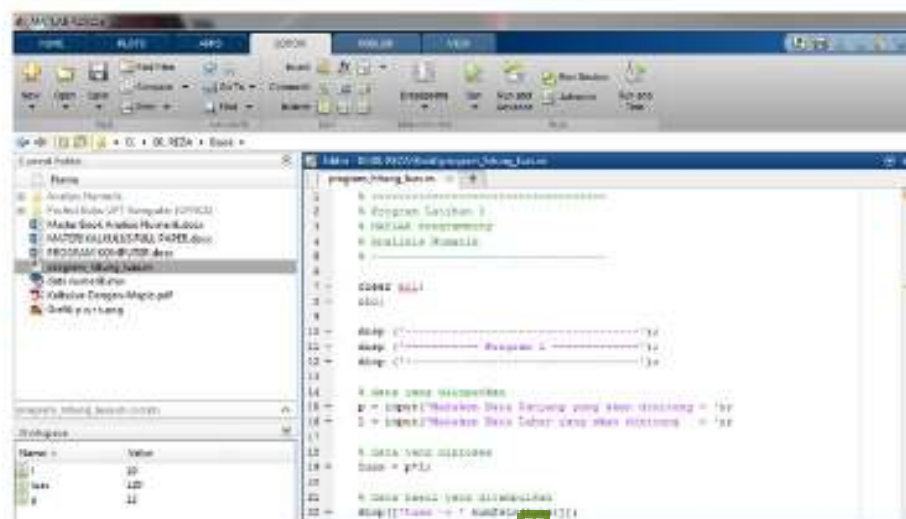
Gambar. 1.6. Tampilan Current Directory MATLAB

Window¹⁷ ini juga fasilitas yang diperkenalkan pada versi 6, berfungsi sebagai browser direktori aktif, yang memiliki fungsi hampir sama dengan window eksplor.



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 programmer.

b) Cara Bekerja dengan MATLAB

Dalam melakukan pekerjaan pemograman 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

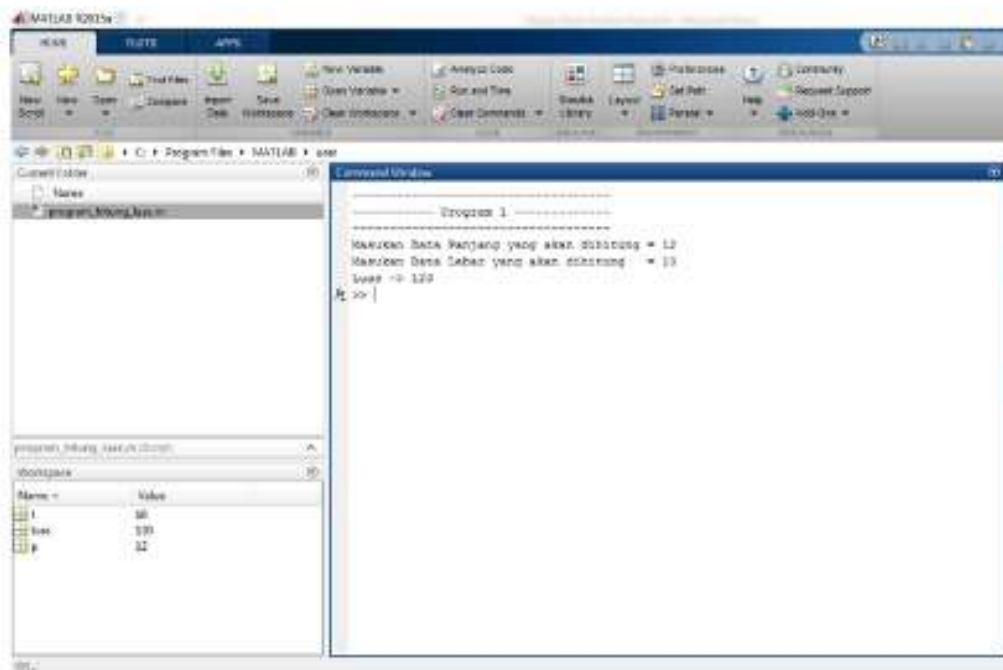
Untuk mengetahui letak pada program penyimpanan di MATLAB, maka perlu diposisikan dahulu pada penyimpanan pada direktori pada bagian current folder di program MATLAB. Kemudian cek isi direktori dengan perintah di bawah ini.

```

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.

21

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 sebelumnya direktori yang didaftarkan pada daftar pencariannya, jika ditemukan eksekusi. Jika tidak MATLAB akan menyampaikan pesan sebagaimana berikut ini.

4. >> kubus

5. ??? 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 di klasifikasikan ke dalam tiga bagian, yaitu Operator Aritmatika, Operator Relasional dan Operator Logika. Adapun penjelasan dari masing-masing operator kami berikan dalam lampiran Variabel dan Operator Dasar MATLAB.

1.4. Pendahuluan Analisis Numerik

2 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.

$$5x^8 - 2.25x^6 - 120x^2 + 15x^3 - 120x^2 - x + 90 = 0$$

34 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 ekstrim fungsi, pertamanya harus menentukan turunan fungsi, dengan menjadikan ruas kanannya sama dengan nol, lalu memeriksa jenis dari titik ekstrimnya. 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 satupun metode yang tersedia untuk menyelesaikan permasalahan-permasalahan jenis ini.

1.5. Analisis Numerik dan Analisis Analitik

Analisis Analitik¹²

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 analitiknya 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 Numerik³⁴

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.

²³

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 aritmatika 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.

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

Melihat persamaan di atas, maka akan menjadi:

$$\sum y_i - a_1 \sum x_i - na_0 = 0 \dots (*)$$

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,

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

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

$$na_0 + \left(\sum x_i\right) a_1 = \sum y_i$$

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

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 D_t 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

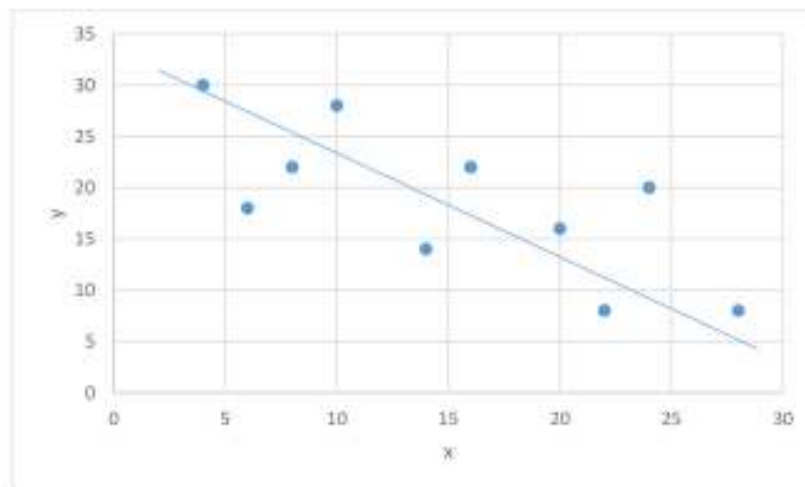
Penyelesaian:

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
Σ	152	186	2432	2912

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

Tabel. 5.2. Tabulasi Data Koefisien Korelasi Regresi Linier

No.	x	y	$x_i \cdot y_i$	x_i^2	D_t	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
Σ	152	186	2432	2912	496,40	236,79

Maka hasil dari nilai D_t 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:

$$a_0 = 28,5849 \text{ dan } a_1 = -0,6569$$

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:

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

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

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

Sehingga jumlah dari kuadrat diberikan oleh:

$$S = \sum (y_i - \hat{y}_i)^2 = \sum (y_i - a_0 - a_1x - a_2x^2)^2$$

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:

$$na_0 + a_1 \sum x_i + a_2 \sum x_i^2 = \sum y_i$$

$$a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 = \sum x_i y_i$$

$$a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 = \sum x_i^2 y_i$$

Maka diperoleh pendekatan $\hat{y} = a_0 + a_1x + a_2x^2$ 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	$x_i \cdot y_i$	$x_i^2 \cdot y_i$
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.

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

$$\hat{y} = 4,821429 + 1,057143 x - 0,178571 x^2$$

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 seksama 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.

```

1  % interpolasi_linier --- Program analisis numerik
2  %
3  % --- interpolasi linier dengan dua data diketahui
4  % --- dengan penghitungan galat mutlak hampiran
5  % --- dengan tingkat ketelitian 0,001
6  %
7  %
8  % author : reza setyandak
9  %
10 %
11 % input data
12 %
13 % Masukan data yang diketahui
14 %
15 % input('Nilai DATA x yang dihitung = ');
16 %
17 % x0=input('Nilai DATA 1 yang diketahui = ');
18 %
19 % x1=input('Nilai DATA 2 yang diketahui = ');
20 %
21 % fx0=input('Hasil DATA 1 yang diketahui = ');
22 %
23 % fx1=input('Hasil DATA 2 yang diketahui = ');
24 %
25 % Masukan hasil data yang diketahui
26 %
27 % fxx=input('Nilai DATA x yang dihitung = ');
28 %
29 %
30 %
31 % proses data interpolasi linier
32 %
33 % fxx=((fx0)+((fx1-fx0)/(x1-x0))*(x-x0));
34 %
35 % galat_mutlak=(fxx-fx);
36 %
37 % galat_mutlak_hampiran=((galat_mutlak)/fxx)*100;
38 %
39 %
40 %
41 %
42 %
43 %
44 %
45 %
46 %
47 %
48 %
49 %
50 %
51 %
52 %
53 %
54 %
55 %
56 %
57 %
58 %
59 %
60 %
61 %
62 %
63 %
64 %
65 %
66 %
67 %
68 %
69 %
70 %
71 %
72 %
73 %
74 %
75 %
76 %
77 %
78 %
79 %
80 %
81 %
82 %
83 %
84 %
85 %
86 %
87 %
88 %
89 %
90 %
91 %
92 %
93 %
94 %
95 %
96 %
97 %
98 %
99 %
100 %

```

Gambar. 5.5. Tampilan file-m dari MATLAB untuk masalah interpolasi linier

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

```

Command Window
>> interpolasi_linier
interpolasi_linier --- Program analisis numerik
                    --- interpolasi linier dengan dua data diketahui
                    --- dengan penghitungan galat mutlak hampiran
                    --- dengan tingkat ketelitian 0.001

=====
author : reza setyanah
=====

Masukkan data yang diketahui
Nilai DATA x yang dihitung =9.2
Nilai DATA 1 yang diketahui =9.0
Nilai DATA 2 yang diketahui =9.6
Hasil DATA 1 yang diketahui =2.1972
Hasil DATA 2 yang diketahui =2.2813
Masukkan hasil data yang diketahui
Nilai DATA x yang dihitung =2.2192

fx =

    2.2188

=====
HASIL PERHITUNGAN INTERPOLASI
=====
Interpolasi Linier f(x) yang dicari =    2.219
Galat Mutlak Hampiran    =    0.016
fx >> |

```

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 seksama langkah penyelesaian interpolasi linier dengan menggunakan program MATLAB berikut.

7 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 setyansah
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. $\text{disp('=====')};$

36. $\text{disp('Hasil data interpolasi lagrange orde 3')};$

37. $\text{disp('=====')};$

38. $\text{fprintf('nilai hasil L0 = %.4f\n',L0)};$

39. $\text{fprintf('nilai hasil L1 = %.4f\n',L1)};$

40. $\text{fprintf('nilai hasil L2 = %.4f\n',L2)};$

41. $\text{fprintf('nilai hasil L3 = %.4f\n',L3)};$

42. $\text{fprintf('nilai interpolasi = %.4f\n',hasil)};$

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.

```

1 % lagrange -- Program analisis numerik
2 %           -- interpolasi lagrange dengan empat data diketahui
3 %           -- interpolasi orde-3 dengan tingkat ketelitian 0.0001
4
5 % -----
6 % author : resa setyansah
7 % -----
8
9 clear; help lagrange;
10 disp('-----');
11 disp('Masukkan data yang diketahui');
12 disp('-----');
13 %input data
14 x=input('diketahui nilai x      = ');
15 x0=input('diketahui nilai x0    = ');
16 x1=input('diketahui nilai x1    = ');
17 x2=input('diketahui nilai x2    = ');
18 x3=input('diketahui nilai x3    = ');
19 fx0=input('diketahui nilai f(x0) = ');
20 fx1=input('diketahui nilai f(x1) = ');
21 fx2=input('diketahui nilai f(x2) = ');
22 fx3=input('diketahui nilai f(x3) = ');
23
24 % -----
25 % proses data
26 % -----
27 L0=(x-x1)/(x0-x1)*(x-x2)/(x0-x2)*(x-x3)/(x0-x3);
28 L1=(x-x0)/(x1-x0)*(x-x2)/(x1-x2)*(x-x3)/(x1-x3);
29 L2=(x-x0)/(x2-x0)*(x-x1)/(x2-x1)*(x-x3)/(x2-x3);
30 L3=(x-x0)/(x3-x0)*(x-x1)/(x3-x1)*(x-x2)/(x3-x2);
31 hasil=L0*fx0+L1*fx1+L2*fx2+L3*fx3;
32
33 % -----
34 % tampilan data : interpolasi lagrange orde 3
35 % -----
36 disp('-----');
37 disp('Hasil data interpolasi lagrange orde 3');
38 disp('-----');
39 fprintf('nilai hasil L0      = %.4f \n',L0);
40 fprintf('nilai hasil L1      = %.4f \n',L1);
41 fprintf('nilai hasil L2      = %.4f \n',L2);
42 fprintf('nilai hasil L3      = %.4f \n',L3);
43 fprintf('nilai interpolasi = %.4f \n',hasil);

```

Gambar. 5.7. Tampilan file-m dari MATLAB untuk masalah interpolasi lagrange

Penerapan dalam perhitungan pemograman MATLAB dengan perhitungan interpolasi lagrange dengan hasil dalam *command window* sebagai berikut.

```

Command Window
>> lagrange
lagrange -- Program analisis numerik
-- interpolasi lagrange dengan empat data diketahui
-- interpolasi orde-3 dengan tingkat ketelitian 0.0001

=====
Masukkan data yang diketahui
=====
diketahui nilai x      = 4
diketahui nilai x0     = 2
diketahui nilai x1     = 3
diketahui nilai x2     = 6
diketahui nilai x3     = 9
diketahui nilai f(x0)  = 0.6931
diketahui nilai f(x1)  = 1.0986
diketahui nilai f(x2)  = 1.7918
diketahui nilai f(x3)  = 2.1972

=====
Hasil data interpolasi lagrange orde 3
=====
nilai hasil L0        = -0.3571
nilai hasil L1        = 1.1111
nilai hasil L2        = 0.2778
nilai hasil L3        = -0.0317
nilai interpolasi     = 1.4011

fx >>

```

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;
17. xi_kuadrat=xi.^2;
18. nx=length(xi);
19. ny=length(yi);
20. disp('=====');
    '=====');
21. disp('data nilai xi*yi');
22. disp(xiyi);
23. disp('data nilai xi^2');
24. disp(xi_kuadrat);
25. disp('=====');
    '=====');
26. juml_xi=sum(xi);
27. juml_yi=sum(yi);
28. juml_xiyi=sum(xiyi);
29. juml_xi_kuadrat=sum(xi_kuadrat);
30. rerata_x=(juml_xi/nx);
31. rerata_y=(juml_yi/ny);
32. a1=((nx*juml_xiyi)-
    (juml_xi*juml_yi))/((nx*juml_xi_kuadrat)-
    (juml_xi^2));
33. a0=(rerata_y-(a1*rerata_x));
34. disp('=====');
    '=====');

```

```

35. fprintf('persamaan regresi linier = %.4f + %.4f x
    \n',a0,a1);
36. disp('=====');
37. Y_bar=rerata_y;
38. Dt=yi-Y_bar;
39. Dt_kuadrat=Dt.^2;
40. D=((yi)-a0-(xi*a1));
41. D_kuadrat=D.^2;
42. Dt_total=sum(Dt_kuadrat);
43. D_total=sum(D_kuadrat);
44. r=sqrt((Dt_total-D_total)/Dt_total);
45. disp('=====');
46. fprintf('nilai r = %.4f\n',r);
47. disp('=====');
48. disp('=====KATE
    GORI=====');
49. if (r > 0)
50.     disp('baik dalam menyesuaikan interval');
51. else
52.     if d==1
53.         disp('sempurna');
54.     else
55.         disp('jelek');
56.     end
57. end
58. disp('=====KETER
    ANGAN=====');
    );

```

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.

```

1 % regresi --- Program analisis statistik
2 % --- menghitung interpolasi dan regresi linier
3 % --- dengan tingkat ketelitian 0.0001
4 %
5 % author : reza betjansah
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;
17 - xi_kuadrat=xi.^2;
18 - nx=length(xi);
19 - ny=length(yi);
20 - disp('-----');
21 - disp('data nilai xi*yi:');
22 - disp(xiyi);
23 - disp('data nilai xi^2:');
24 - disp(xi_kuadrat);
25 - disp('-----');
26 - jumlah_xi=sum(xi);
27 - jumlah_yi=sum(yi);
28 - jumlah_xiyi=sum(xiyi);
29 - jumlah_xi_kuadrat=sum(xi_kuadrat);
30 - rerata_x=(jumlah_xi/nx);

```

```

31 - rerata_y=(juml_y/ny);
32 - a1=(nm*juml_xiyi)-(juml_xi*juml_yi)/((nm*juml_xi_kuadrat)-(juml_xi^2));
33 - a0=(rerata_y-(a1*rerata_x));
34 - disp('=====');
35 - fprintf('persamaan regresi linier = %4f + %4f x \n',a0,a1);
36 - disp('=====');
37 - Y_bar=rerata_y;
38 - Dc=yi-Y_bar;
39 - Dc_kuadrat=Dc.^2;
40 - D=(yi-a0-(xi*a1));
41 - D_kuadrat=D.^2;
42 - Dc_total=sum(Dc_kuadrat);
43 - D_total=sum(D_kuadrat);
44 - r=sqrt((Dc_total-D_total)/Dc_total);
45 - disp('=====');
46 - fprintf('nilai r = %4f \n',r);
47 - disp('=====');
48 - disp('=====KATEGORI=====');
49 - if (r > 0)
50 -     disp('Data dalam besertuaian interval');
51 - else
52 -     if d==1
53 -         disp('sempurna');
54 -     else
55 -         disp('jelek');
56 -     end
57 - end
58 - disp('=====PETERAJIAN=====');
59 - disp('jika r = 0, kategori jelek');
60 - disp('jika r = 1, kategori sempurna');

```

Gambar. 5.9. Tampilan m-file dari MATLAB untuk masalah regresi linier

Penerapan dalam perhitungan pemograman MATLAB dengan perhitungan regresi linier dengan hasil dalam *command window* sebagai berikut.

```

Command Window

author : reza setyansah
=====
DATA YANG DIKETASUI
=====
Masukkan nilai x [dalam array] =[4 6 8 10 14 16 20 22 24 28]
Masukkan nilai y [dalam array] =[30 18 22 28 14 22 16 8 20 8]
=====
data nilai xi*yi
    120    108    176    280    196    352    320    176    480    224

data nilai xi^2
     16     36     64    100    196    256    400    484    576    784

=====
persamaan regresi linier = 28.5851 + -0.6569 x
=====
nilai r = 0.7232
=====
-----KATEGORI-----
baik dalam menyesuaikan interval
-----KETERANGAN-----
jika r = 0, kategori jelek
jika r = 1, kategori sempurna
fx >> |

```

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


```

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;
17.    xi_kuadrat=xi.^2;
18.    xi2yi=xi_kuadrat.*yi;
19.    xi_triple=xi.^3;
20.    xi_quad=xi.^4;
21.    nx=length(xi);
22.    ny=length(yi);
23.    disp('=====Tam
      pilan data=====');
24.    disp('data nilai xi*yi');
25.    disp(xiyi);
26.    disp('data nilai xi^2*yi');
27.    disp(xi2yi);
28.    disp('data nilai xi^2');
29.    disp(xi_kuadrat);
30.    disp('data nilai xi^3');
31.    disp(xi_triple);
32.    disp('data nilai xi^4');
33.    disp(xi_quad);
34.    disp('=====')
      =====
      =');
35.    juml_xi=sum(xi);
36.    juml_yi=sum(yi);
37.    juml_xiyi=sum(xiyi);

```

```

38.    juml_xi2yi=sum(xi2yi);
39.    juml_xi_kuadrat=sum(xi_kuadrat);
40.    juml_xi_triple=sum(xi_triple);
41.    juml_xi_quad=sum(xi_quad);
42.    rerata_x=(juml_xi/nx);
43.    rerata_y=(juml_yi/ny);
44.    disp('=====Tam
    pilan data=====');
45.    fprintf('Jumlah data xi dan yi = %.0f dan %.0f
    \n',nx,ny);
46.    fprintf('Jumlah data xi      = %.4f\n',juml_xi);
47.    fprintf('Jumlah data xi^2      = %.4f
    \n',juml_xi_kuadrat);
48.    fprintf('Jumlah data xi^3      = %.4f
    \n',juml_xi_triple);
49.    fprintf('Jumlah data xi^4      = %.4f
    \n',juml_xi_quad);
50.    fprintf('Jumlah data yi      = %.4f\n',juml_yi);
51.    fprintf('Jumlah data xi*yi      = %.4f
    \n',juml_xiyi);
52.    fprintf('Jumlah data xi^2*yi      = %.4f
    \n',juml_xi2yi);
53.    %
    =====proses
    data=====
54.    A = [nx  juml_xi  juml_xi_kuadrat;juml_xi
    juml_xi_kuadrat  juml_xi_triple;juml_xi_kuadrat
    juml_xi_triple juml_xi_quad];
55.    b = [juml_yi;juml_xiyi;juml_xi2yi];
56.    det(A);
57.    x=inv(A)*b;

```

```

58.    disp('===== hasil data
      a0, a1 dan a2 =====');
59.    disp(x);
60.    disp('=====
      =====
      =');
61.    disp('Masukkan hasil data a0, a1 dan a2 untuk
      membuat persamaan regresi');
62.    a0=input('Masukkan nilai a0 =');
63.    a1=input('Masukkan nilai a1 =');
64.    a2=input('Masukkan nilai a2 =');
65.    fprintf('persamaan regresi polinomial y^ = %.4f
      + %.4f x + %.4f x^2 \n',a0,a1,a2);
66.    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.

```

1 % regresi: --- Program analisis statistik
2 % --- menggunakan matriks untuk regresi polinomial
3 % --- dimensi vektor hasil: 3.0001
4 %
5 % author : zeta setyawan
6 %
7 % clear help regresi_polinomial;
8 %
9 % input data
10 % disp('=====')
11 % disp('DATA YANG DIKIRIM:'))
12 % disp('=====')
13 % xl=input('Masukkan nilai x (dalam array) :');
14 % yl=input('Masukkan nilai y (dalam array) :');
15 % proses data nilai
16 % xi=x(:);
17 % xi_kuadrat=xi.^2;
18 % xi_y=xi.*kuadrat;
19 % xi_triple=xi.^3;
20 % xi_quad=xi.^4;
21 % n=length(xi);
22 % ny=length(yl);
23 % disp('=====')
24 % disp('Data nilai xi:');
25 % disp(xi);
26 % disp('Data nilai xi^2:');
27 % disp(xi_kuadrat);
28 % disp('Data nilai xi^3:');
29 % disp(xi_triple);
30 % disp('Data nilai xi^4:');
31 % disp(xi_quad);
32 % disp('Data nilai xi^5:');
33 % disp(xi_quad);
34 % disp('=====')
35 % jumlah_xi=sum(xi);
36 % jumlah_y=sum(yl);
37 % jumlah_xi_y=sum(xi_y);
38 % jumlah_xi_y_kuadrat=sum(xi_kuadrat);
39 % jumlah_xi_triple=sum(xi_triple);
40 % jumlah_xi_quad=sum(xi_quad);
41 % rerata_x=(jumlah_xi/n);
42 % rerata_y=(jumlah_y/ny);
43 % disp('=====')
44 % disp('Tampilkan data:');
45 % fprintf('Jumlah data xi dan yl = %d dan %d \n',n,ny);
46 % fprintf('Jumlah data xi = %d \n',jumlah_xi);
47 % fprintf('Jumlah data xi^2 = %d \n',jumlah_xi_kuadrat);
48 % fprintf('Jumlah data xi^3 = %d \n',jumlah_xi_triple);
49 % fprintf('Jumlah data xi^4 = %d \n',jumlah_xi_quad);
50 % fprintf('Jumlah data yl = %d \n',jumlah_yl);
51 % fprintf('Jumlah data xi*yl = %d \n',jumlah_xi_y);
52 % fprintf('Jumlah data xi^2*yl = %d \n',jumlah_xi_y_kuadrat);
53 %
54 %
55 % h = [jumlah_xi jumlah_xi_kuadrat;jumlah_xi_triple;jumlah_xi_quad;jumlah_xi_y;jumlah_xi_y_kuadrat];
56 % b = [jumlah_y;jumlah_xi_y;jumlah_xi_y_kuadrat];
57 % det(h);
58 % x=inv(h)*b;
59 % disp('=====')
60 % disp('=====')
61 % disp('=====')
62 % disp('=====')
63 % disp('=====')
64 % disp('=====')
65 % disp('=====')
66 % disp('=====')
67 % disp('=====')
68 % disp('=====')
69 % disp('=====')
70 % disp('=====')
71 % disp('=====')
72 % disp('=====')
73 % disp('=====')
74 % disp('=====')
75 % disp('=====')
76 % disp('=====')
77 % disp('=====')
78 % disp('=====')
79 % disp('=====')
80 % disp('=====')
81 % disp('=====')
82 % disp('=====')
83 % disp('=====')
84 % disp('=====')
85 % disp('=====')
86 % disp('=====')
87 % disp('=====')
88 % disp('=====')
89 % disp('=====')
90 % disp('=====')
91 % disp('=====')
92 % disp('=====')
93 % disp('=====')
94 % disp('=====')
95 % disp('=====')
96 % disp('=====')
97 % disp('=====')
98 % disp('=====')
99 % disp('=====')
100 % disp('=====')

```

Gambar. 5.11. Tampilan m-file dari MATLAB untuk masalah regresi polinomial

Penerapan dalam perhitungan pemograman MATLAB dengan perhitungan regresi polinomial dengan hasil dalam *command window* sebagai berikut.

```

Command Window

>> regresi_polinomial
regresi --- Program analisis numerik
        --- menghitung interpolasi dan regresi polinomial
        --- dengan tingkat ketelitian 0.0001
=====
author : riza setyanah
=====

DATA YANG DIKETAHUI
=====
Masukkan nilai x (dalam array) =[0 2 3 4 5 6]
Masukkan nilai y (dalam array) =[5 5 8 6 5 5]
=====Tampilan data=====
data nilai xi*yi
      0    10    24    24    25    30

data nilai xi^2*yi
      0    20    72    96    125    180

data nilai xi^2
      0     4     9    16    25    36

data nilai xi^3
      0     8    27    64    125    216

data nilai xi^4
      0     16    81    256    625    1296
=====Tampilan data=====
Jumlah data xi dan yi = 6 dan 6
Jumlah data xi        = 20.0000
Jumlah data xi^2      = 90.0000
Jumlah data xi^3      = 440.0000
Jumlah data xi^4      = 2274.0000
Jumlah data yi        = 34.0000
Jumlah data xi*yi     = 113.0000
Jumlah data xi^2*yi   = 493.0000
===== Hasil data a0, a1 dan a2 =====
      4.8214
      1.0571
     -0.17857

=====
Masukkan hasil data a0, a1 dan a2 untuk membuat persamaan regresi
Masukkan nilai a0 =4.8214
Masukkan nilai a1 =1.0571
Masukkan nilai a2 =-0.17857
persamaan regresi polinomial y' = 4.8214 + 1.0571 x + -0.1786 x^2
=====

```

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.

$$\hat{y} = 4,821429 + 1,057143 x - 0,178571 x^2$$

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 $f(x)$ 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.

$$\int_0^2 \frac{2 + \cos(1 + x^{\frac{3}{2}})}{\sqrt{1 + 0.5 \sin x}} e^{0.5x} dx$$

- (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

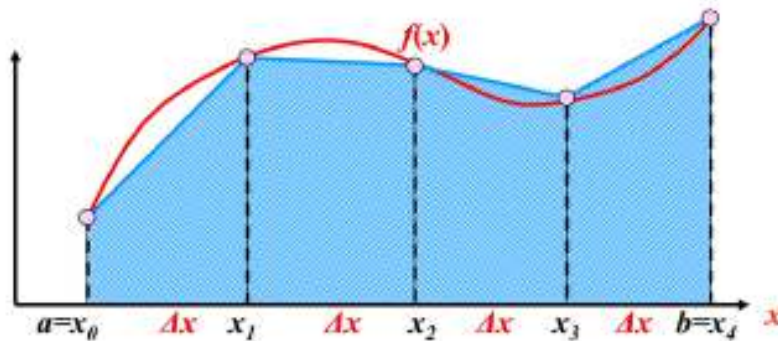
dengan menggunakan persamaan polinomial order tiga yang melalui empat titik.

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.

$$I = \int_0^4 e^x dx = [e^x]_0^4 = (e^4 - e^0) = 53,598150$$

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:

$$I = \frac{\Delta x}{2} [f(a) + f(b) + 2 \sum_{j=1}^{n-1} f(x_j)]$$

$$I = \frac{1}{2} [e^0 + e^4 + 2(e^1 + e^2 + e^3)] = 57,991950$$

Kesalahan relatif terhadap nilai eksak:

$$\varepsilon = E_R = \frac{|53,598150 - 57,991950|}{53,598150} \times 100\% = 8,2\%$$

Apabila digunakan metode trapesium dengan koreksi ujung, maka integral dihitung sebagai berikut:

$$I = \frac{\Delta x}{2} [f(a) + f(b) + 2 \sum_{j=1}^{n-1} f(x_j)] - \frac{\Delta x^2}{12} [f''(b) - f''(a)]$$

$$I = \frac{1}{2} [e^0 + e^4 + 2(e^1 + e^2 + e^3)] - \frac{1}{12} (e^4 - e^0)$$

$$I = 57,991950 - 4,466513$$

$$I = 53,525437$$

Kesalahan/galat relatif terhadap nilai eksak

$$\varepsilon = E_R = \frac{53,598150 - 53,525437}{53,598150} \times 100\% = 0,14\%$$

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.

$$I = \frac{\Delta x}{2} [f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_i)] - \frac{\Delta x^2}{12} [f'(b) - f'(a)]$$

$$f'(x_1 = a = 0) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{f(1) - f(0)}{1 - 0} = \frac{3 - 1}{1} = 2$$

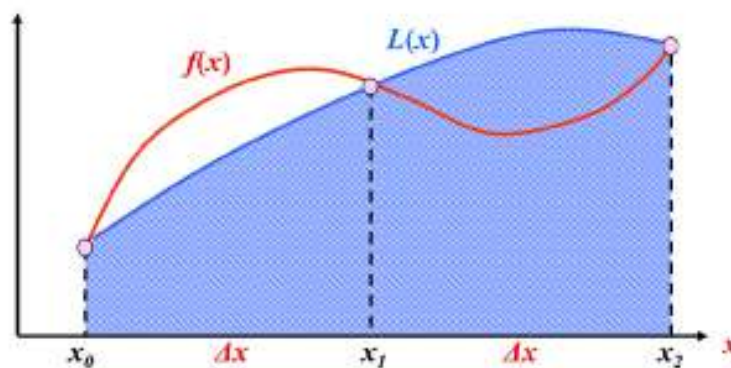
$$f'(x_n = b = 4) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} = \frac{f(4) - f(3)}{4 - 3} = \frac{33 - 19}{1} = 14$$

$$I = \frac{1}{2} [1 + 33 + 2(3 + 9 + 19)] - \frac{1}{12} (14 - 2) = 48 - 1 = 47$$

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 ⁹ 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:

$$I = \int_0^4 e^x dx = [e^x]_0^4 = (e^4 - e^0) = 53,598150$$

Penghitungan dengan aturan simpson 1/3:

$$A_1 = \frac{b-a}{6} [f(a) + 4f(c) + f(b)] = \frac{4-0}{6} [e^0 + 4e^2 + e^4] = 56,7696$$

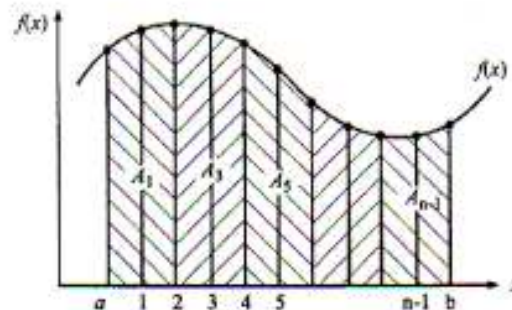
Untuk mengetahui tingkat ketelitian, dibandingkan hasil hitungan numerik dan analitis:

$$\varepsilon = \frac{53,598750 - 56,7696}{53,598750} \times 100\% = 5,917\%$$

Terlihat bahwa hasil menggunakan metode aturan ²⁸ 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,

$$\int_a^b f(x) dx = A_1 + A_3 + \dots + A_{n-1}$$

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

$$\int_a^b f(x) dx = \frac{\Delta x}{3} (f_0 + 4f_1 + f_2) + \frac{\Delta x}{3} (f_2 + 4f_3 + f_4) + \dots + \frac{\Delta x}{3} (f_{n-2} + 4f_{n-1} + f_n)$$

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 ($i=1,3,5,\dots$)

dan untuk

$$2 \sum_{i=2}^{n-2} f(x_i)$$

adalah untuk nilai i genap ($i=2,4,6,\dots$)

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:

$$\varepsilon = E_R = \frac{53,598150 - 53,863846}{53,598150} \times 100\% = 0,5\%$$

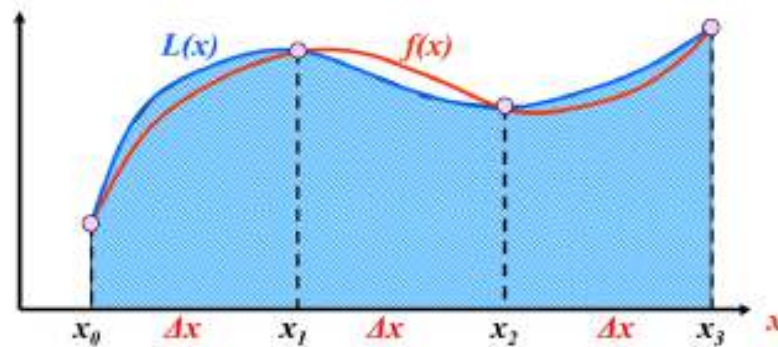
Terlihat bahwa hasil menggunakan metode aturan simpson 1/3 dengan panjang pias berarti memberikan hasil lebih baik dari metode simpson.

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:

$$I = \frac{3\Delta x}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

dengan

$$\Delta x = \frac{b-a}{3}$$

Jadi, metode Simpson 3/8 dapat juga ditulis dalam bentuk:

$$I = (b-a) \frac{[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]}{8}$$

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.

$$I = (b-a) \frac{[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]}{8}$$

$$I = (4-0) \frac{[e^0 + 3e^{1,3333} + 3e^{2,6667} + e^4]}{8} = 55,07798$$

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

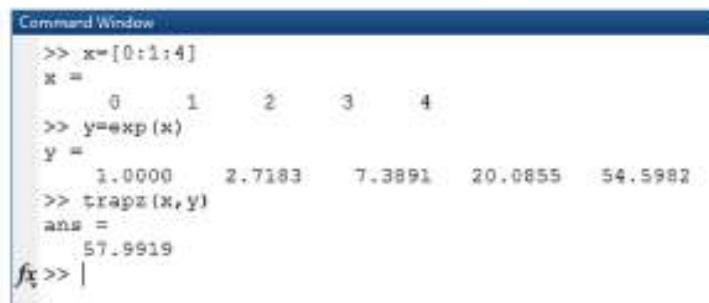
Adapun perhitungan simulasi MATLAB pada metode integrasi trapezium, menyelesaikan permasalahan data berikut. Mengacu dari permasalahan yang dikembangkan dari contoh, perhatikan seksama langkah penyelesaian metode integrasi trapezium dengan menggunakan program MATLAB berikut.

$$I = \int_0^4 e^x dx$$

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. $\text{trapz}(x,y)$

Penerapan dalam perhitungan pemograman MATLAB dengan perhitungan integrasi trapezium dengan hasil dalam command window sebagai berikut.



```

Command Window
>> x=[0:1:4]
x =
    0    1    2    3    4
>> y=exp(x)
y =
    1.0000    2.7183    7.3891   20.0855   54.5982
>> trapz(x,y)
ans =
    57.9919
fx>> |
  
```

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, menunjukan 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 <https://www.mathworks.com/matlabcentral/fileexchange/25255-simpson> untuk menyelesaikan data tersebut sebagai berikut.

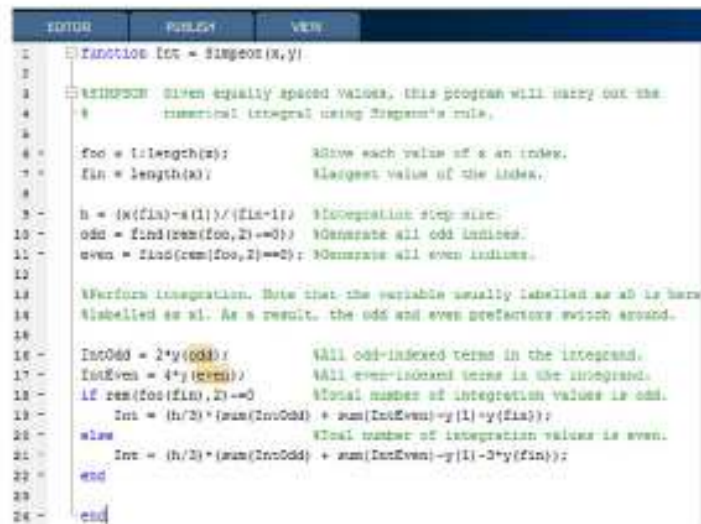
1. function Int = Simpson(x,y)
- 2.
3. %SIMPSON Given equally spaced values, this program will carry out the
4. % numerical integral using Simpson's rule.
- 5.
6. foo = 1:length(x); %Give each value of x an index.
7. fin = length(x); %Largest value of the index.
- 8.
9. h = (x(fin)-x(1))/(fin-1); %Integration step size.
10. odd = find(rem(foo,2)~=0); %Generate all odd indices.
11. even = find(rem(foo,2)==0); %Generate all even indices.
- 12.
13. %Perform integration. Note that the variable usually labelled as x0 is here
14. %labelled as x1. As a result, the odd and even prefactors switch around.

```

15.
16.    IntOdd = 2*y(odd);           %All odd-indexed
    terms in the integrand.
17.    IntEven = 4*y(even);        %All even-indexed
    terms in the integrand.
18.    if rem(foo(fin),2)~=0        %Total number of
    integration values is odd.
19.        Int = (h/3)*(sum(IntOdd) + sum(IntEven)-
    y(1)-y(fin));
20.    else                        %Toal number of integration
    values is even.
21.        Int = (h/3)*(sum(IntOdd) + sum(IntEven)-
    y(1)-3*y(fin));
22.    end
23.
24.    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.



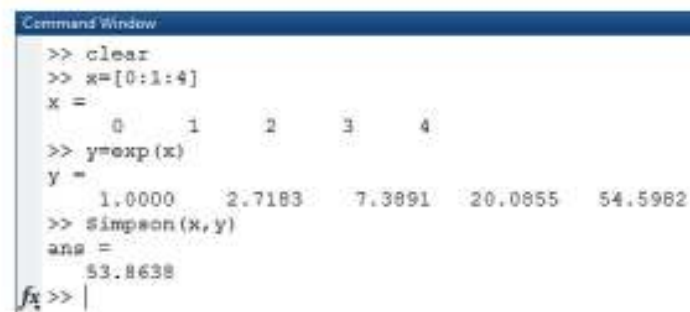
```

1 function Itr = Simpson(x,y)
2
3 %SIMPSON Given equally spaced values, this program will carry out the
4 %numerical integral using Simpson's rule.
5
6 foo = 1:length(x); %Give each value of x an index.
7 fin = length(x); %Largest value of the index.
8
9 h = (x(fin)-x(1))/(fin-1); %Integration step size.
10 odd = find(rem(foo,2)~=0); %Generate all odd indices.
11 even = find(rem(foo,2)==0); %Generate all even indices.
12
13 %Perform integration. Note that the variable usually labelled as n is here
14 %labelled as n1. As a result, the odd and even prefactors switch around.
15
16 IntOdd = 2*y(odd); %All odd-indexed terms in the integrand.
17 IntEven = 4*y(even); %All even-indexed terms in the integrand.
18 if rem(fin,2)~=0 %Total number of integration values is odd.
19 Int = (h/2)*(sum(IntOdd) + sum(IntEven)-y(1)-y(fin));
20 else %Total number of integration values is even.
21 Int = (h/2)*(sum(IntOdd) + sum(IntEven)-y(1)-2*y(fin));
22 end
23
24 end

```

Gambar. 6.6. Tampilan File-M MATLAB Metode Integrasi Simpson

Penerapan dalam perhitungan pemograman MATLAB dengan perhitungan integrasi simpson untuk panjang pias dengan hasil dalam command window sebagai berikut.



```

Command Window
>> clear
>> x=[0:1:4]
x =
    0    1    2    3    4
>> y=exp(x)
y =
    1.0000    2.7183    7.3891   20.0855   54.5982
>> Simpson(x,y)
ans =
    53.8638
fx >> |

```

Gambar. 6.7. Tampilan Command Window MATLAB Metode Integrasi Simpson Panjang Pias

$$I = (4 - 0) \frac{[e^0 + 3e^{1,3333} + 3e^{2,6667} + e^4]}{8} = 55,07798$$

Sedemikian sehingga perolehan hasil di atas, menunjukan 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.

⁶⁴ Chapra Steven C., Canale Raymond P. 1985. Numerical Methods For Engineers. New York. Mc. Graw-Hill Book Company.

⁶³ Hamming, R.W. 1973. Numerical Methods for Scientists and Engineers. Second Edition. New York: Dover Publications, Inc.

<https://www.mathworks.com/>

Tjolleng, Amir. 2017. Pengantar Pemograman MATLAB. Jakarta : PT. Elex Media Komputindo.

Triatmodjo, Bambang. 2002. Metode Numerik. Jakarta: Penerbit Beta Offset

Sahid. 2005. Komputasi Numerik dengan Matlab. Yogyakarta: Penerbit Andi Offset

Suarga. 2014. Komputasi Numerik – Pemograman MATLAB untuk Metoda Numerik. Yogyakarta: Penerbit Andi OFFSET.

Tentang Penulis

Penulis 1 : 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.

Penulis 2 : 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.

Buku Anilisi Numerik

ORIGINALITY REPORT

30%

SIMILARITY INDEX

29%

INTERNET SOURCES

2%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

risqi.blog.com

Internet Source

2%

2

choirudiniaimnumetro.files.wordpress.com

Internet Source

2%

3

fr.slideshare.net

Internet Source

1%

4

www.mathworks.com

Internet Source

1%

5

www.scribd.com

Internet Source

1%

6

vdocuments.site

Internet Source

1%

7

fr.scribd.com

Internet Source

1%

8

etheses.uin-malang.ac.id

Internet Source

1%

9

es.scribd.com

Internet Source

1%

10	repository.usu.ac.id Internet Source	1 %
11	bismaroel.blogspot.com Internet Source	1 %
12	edoc.site Internet Source	1 %
13	pt.scribd.com Internet Source	1 %
14	documents.mx Internet Source	1 %
15	www.slideshare.net Internet Source	1 %
16	ml.scribd.com Internet Source	1 %
17	library.uwp.ac.id Internet Source	1 %
18	alfaruqi.lecturer.pens.ac.id Internet Source	1 %
19	dukunkomputer.org Internet Source	1 %
20	putrihidayani91.blogspot.com Internet Source	1 %
21	inyonoel.blogspot.com Internet Source	1 %

22	irvanzaky.blogspot.com Internet Source	1 %
23	belajarbarengewan.blogspot.com Internet Source	1 %
24	eprints.akakom.ac.id Internet Source	1 %
25	supardi21.wordpress.com Internet Source	<1 %
26	endysibarani17.blogspot.com Internet Source	<1 %
27	Submitted to University of Nottingham Student Paper	<1 %
28	anzdoc.com Internet Source	<1 %
29	de.slideshare.net Internet Source	<1 %
30	irohmunawaroh.blogspot.com Internet Source	<1 %
31	file.upi.edu Internet Source	<1 %
32	Submitted to Texas A&M University, College Station Student Paper	<1 %

33

tiptiktak.com

Internet Source

<1 %

34

slideplayer.info

Internet Source

<1 %

35

amirtjolleng.blogspot.com

Internet Source

<1 %

36

ekowahyuningp.blogspot.com

Internet Source

<1 %

37

yuliana.lecturer.pens.ac.id

Internet Source

<1 %

38

Submitted to iGroup

Student Paper

<1 %

39

ayups.blogspot.com

Internet Source

<1 %

40

maxima-online.org

Internet Source

<1 %

41

id.scribd.com

Internet Source

<1 %

42

journal.unisla.ac.id

Internet Source

<1 %

43

stmik-time.ac.id

Internet Source

<1 %

44

hanifannursyah.blogspot.co.id

Internet Source

<1 %

45

blog.ub.ac.id

Internet Source

<1 %

46

reni.lecturer.pens.ac.id

Internet Source

<1 %

47

www.pmfst.hr

Internet Source

<1 %

48

Naksoo Kim, Byungseok Kang, Seungyoon Lee.
"Prediction and design of edge shape of initial
strip for thick tube roll forming using finite
element method", Journal of Materials
Processing Technology, 2003

Publication

<1 %

49

zenhadi.lecturer.pens.ac.id

Internet Source

<1 %

50

www.coursehero.com

Internet Source

<1 %

51

www.ilmutekniksipilindonesia.com

Internet Source

<1 %

52

Submitted to Higher Education Commission
Pakistan

Student Paper

<1 %

53

www.slideserve.com

Internet Source

<1 %

54	pascalmatlab.blogspot.com Internet Source	<1 %
55	www.docstoc.com Internet Source	<1 %
56	www.mathcs.citadel.edu Internet Source	<1 %
57	Submitted to Universitas Negeri Jakarta Student Paper	<1 %
58	docplayer.info Internet Source	<1 %
59	docslide.net Internet Source	<1 %
60	sutanto.staff.uns.ac.id Internet Source	<1 %
61	eprints.uns.ac.id Internet Source	<1 %
62	zai.lecturer.pens.ac.id Internet Source	<1 %
63	Sanchez, . "Bibliography", Software Solutions for Engineers and Scientists, 2007. Publication	<1 %
64	elka.eepis-its.edu Internet Source	<1 %

Exclude quotes On

Exclude matches < 15 words

Exclude bibliography On