PRAKTIKUM 1
MODELING DAN ANALISIS KESALAHAN

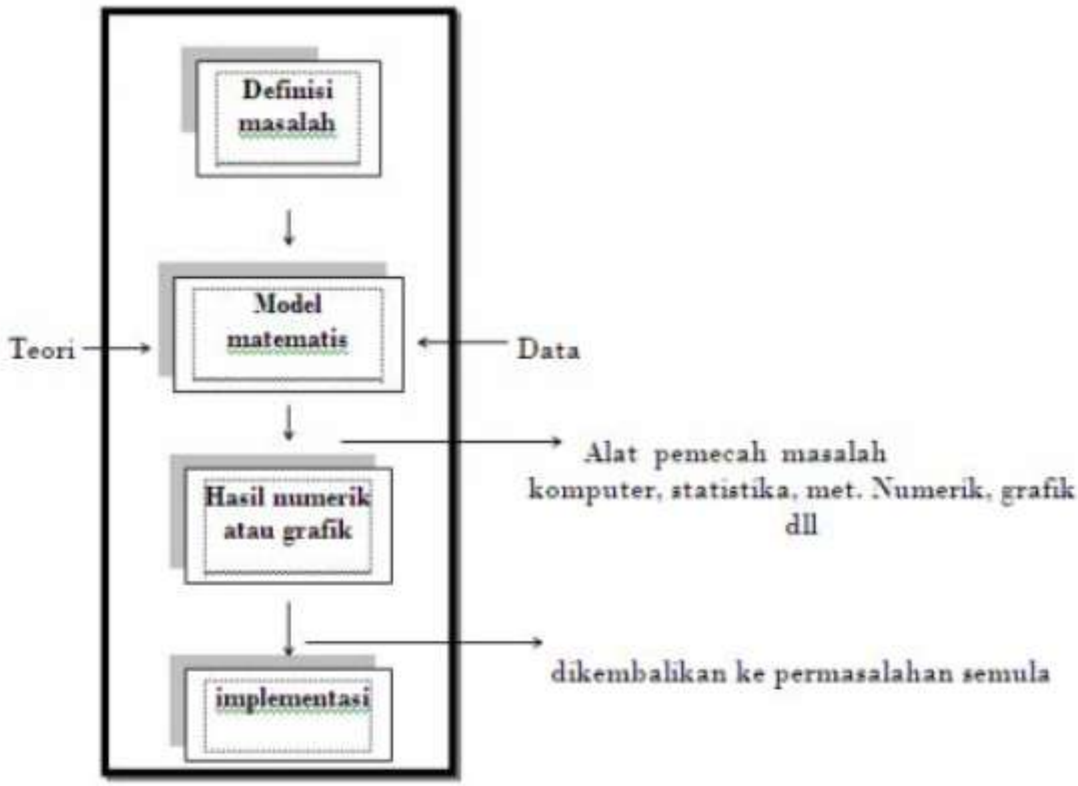
A. TUJUAN PEMBELAJARAN

1. Model Matematika
2. Memahami Deret Taylor
3. Memahami Galat
4. Memahami algoritma dan pembacaan flowchart

B. DASAR TEORI

1. Model Matematika

Model dibuat untuk memudahkan orang dalam menganalisis suatu permasalahan, disamping untuk menghemat waktu, biaya, dan juga mengurangi resiko. Dengan adanya sistem komputer yang demikian canggih saat ini, maka pemodelan ini menjadi lebih mudah dan nyaman dilakukan. Dari sini lahirlah simulasi yang menggunakan komputer untuk menirukan hal-hal yang ada di dunia nyata, yang dapat dianalisis, dievaluasi dan didapatkan hasilnya, serta dapat diulangi kapanpun dengan hasil yang sama. Pemodelan matematik diperlukan untuk membantu menyelesaikan permasalahan rekayasa (permasalahan riil). Gambaran tahapanpemrosesan masalah rekayasa yang secara analitis sulit diselesaikan selanjutnya dibawa ke bentuk model matematik dan diselesaikan secara matematis, aljabar atau statistik dan komputasi. Apabila telah diperoleh penyelesaian matematik proses selanjutnya mengimplementasikan hasil matematis ke masalah rekayasa sbb:



Gambar 1.1 Skema model Matematika
(sumber "Komputasi numeric dengan mathlab",Ardi P)

2. Memahami Deret Taylor

Metode numeric digunakan untuk menyelesaikan persoalan dimana perhitungan secara analitik tidak dapat digunakan. Metode numeric ini berangkat dari pemikiran bahwa permasalahan dapat diselesaikan dengan menggunakan pendekatan-pendekatan yang dapat dipertanggung-jawabkan secara analitik. Metode numerik ini disajikan dalam bentuk algoritma-algoritma yang dapat dihitung secara cepat dan mudah.Pendekatan yang digunakan dalam metode numerik merupakan pendekatan analisis matematis. Sehingga dasar pemikirannya tidak keluar jauh dari dasar pemikiran analitis, hanya saja pemakaian grafis dan teknik perhitungan yang mudah merupakan pertimbangan dalam pemakaian metode numerik.

Pada umumnya fungsi-fungsi yang bentuknya kompleks dapat disederhanakan menjadi fungsi hampiran dalam bentuk fungsi polinomial yang lebih sederhana. Fungsi polinomial lebih mudah dipahami kelakuannya. Apabila kita melakukan pekerjaan hitungan dengan menggunakan fungsi yang sesungguhnya, maka akan kita dapatkan hasil solusi eksak (solusi sejati). Tetapi bila kita melakukan pekerjaan hitungan dengan menggunakan fungsi hampiran, maka akan kita dapatkan hasil solusi hampiran (solusi pendekatan). Perbedaan antara solusi eksak dan solusi hampiran terletak pada adanya galat pada solusi hampiran. Galat pada solusi numerik harus dihubungkan dengan seberapa teliti polinomial dalam menghampiri fungsi yang sesungguhnya. Biasanya dalam menghampiri fungsi yang sesungguhnya, orang menggunakan apa yang disebut dengan deret Taylor.

Deret Taylor biasanya digunakan untuk kuantifikasi atau perkiraan besar kesalahan. Tanpa melalui pembuktian Matematika deret Taylor dapat dirumuskan :

$$f(x) = f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \frac{(x-x_0)}{2!} f''(x_0) + \ldots + \frac{(x-x_0)^m}{m!} f^m(x_0) + \ldots.$$

Andaikan f dan semua turunannya f', f'', f''', …, kontinu di dalam interval [a,b], misalkan x_0 € [a,b], maka untuk nilai disekitar x_0 dan x € [a,b], f(x) didalam deret Taylor:

Contoh:

Hampiri fungsi f(x) = cos(x) ke dalam deret Taylor disekitar $x_0 = 2$

Penyelesaian:
Menentukan turunan cos(x) terlebih dahulu
f(x) = cos(x)
f'(x) = -sin(x)
f''(x) = -cos(x)
f'''(x) = sin(x), dan seterusnya
berdasarkan persamaan diatas, sin(x) dihampiri dengan deret Taylor:

$$\cos(x) = \cos(2) + \frac{(x-2)}{1!}(-\sin(2)) + \frac{(x-2)^2}{2!}(-\cos(2)) + \frac{(x-2)^3}{3!}(\sin(2)) + \ldots.$$

Dimisalkan x – 2 = m, maka:

$$\cos(x) = \cos(2) - m\sin(2) - \frac{m^2}{2}\cos(2) + \frac{m^3}{6}\sin(2) + \ldots.$$

$$\cos(x) = 9{,}9939 - 3{,}4899m - 4{,}9968m^2 + 1{,}7449m^3 + \ldots.$$

Mengingat bahwa algoritma yang dikembangkan dalam metode numerik adalah algoritma pendekatan maka dalam algoritma tersebut akan muncul istilah *iterasi* yaitu pengulangan proses perhitungan. Dengan kata lain perhitungan dalam metode numerik adalah perhitungan yang dilakukan secara berulang-ulang untuk terus-menerus diperoleh hasil yang mana mendekati nilai penyelesaian exact. Perhatikan salah bentuk formulasi dalam metode numeric adalah:

$$x_n = x_{n-1} + \delta x_{n-1}$$

Terlihat bahwa hasil iterasi ke n adalah hasil iterasi ke n-1 (sebelumnya) dengan ditambah δx_{n-1} yang merupakan nilai perbaikan. Sehingga dapat dikatakan bahwa semakain banyak iterasi yang digunakan, maka nilainya semakin mendekati nilai exact atau semakin baik hasil yang diperoleh.

Dengan menggunakan metode pendekatan semacam ini, tentukan setiap nilai hasil perhitungan akan mempunyai *nilai error* (nilai kesalahan/galat). Dalam analisa metode numeric, kesalahan ini menjadi penting artinya. Karena kesalahan dalam pemakaian algoritma pendekatan akan menyebabkan nilai kesalahan yang besar, tentunya ini tidak diharapkan. Sehingga pendekatan metode analitik selalu membahas tingkat kesalahan dan tingkat kecepatan proses yang akan terjadi.

3. Analisa Kesalahan

Kesalahan numerik adalah kesalahan yang timbul karena adanya proses pendekatan.

Hubungan kesalahan dan penyelesaian $\hat{x} = x + e$ adalah nilai yang sebenarnya (nilai eksak)

x adalah nilai pendekatan yang dihasilkan dari metode numerik

e adalah kesalahan numerik.

prosentase antara kesalahan

- Kesalahan fraksional adalah prosentase antara kesalahandan nilai sebenarnya

$$\in = \left(\frac{e}{\hat{x}} \right) x100\%$$

- Pada banyak permasalahan kesalahan fraksional di atas sulit atau tidak bisa dihitung, karena nilai eksaknya tidak diketahui.
- Sehingga kesalahan fraksional dihitung berdasarkan nilai pendekatan yang diperoleh:
- Perhitungan kesalahan semacam ini dilakukan untuk mencapai keadaan konvergensi pada suatu proses iterasi.

$$\in = \left(\frac{x_n - x_{n-1}}{x_n} \right) \times 100\% \qquad \in = \left(\frac{e}{x} \right) \times 100\%$$

C. TUGAS PENDAHULUAN

Carilah contoh permasalahan yang menggunakan solusi eksak dan solusi numerik.

D. PERCOBAAN

1. Bentuklah deret Taylor untuk fungsi f(x) = ln (x) disekitar x_0 = 1
2. pi=3.14159265358979 dan pola 22/7. Tentukan galat berhubungan dengan 4 angka dibelakang koma.
3. Buatlah flowchat dari kasus menentukan bilangan terkecil dari 3 bilangan.

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas.

PRAKTIKUM 2

PENYELESEIAN PERASAMAAN NONLINIER PART 1

A. TUJUAN PEMBELAJARAN

1. Dapat memodelkan fungsi menggunakan gnuplot
2. Dapat melakukan penggantian range setelah melakukan plotting sebuah fungsi menggunakan gnuplot
3. Dapat melakukan scaling untuk sebuah fungsi

B. DASAR TEORI

Penyelesaian persamaan non linier adalah penentuan akar-akar persamaan non linier.Dimana akar sebuah persamaan f(x) =0 adalah nilai-nilai x yang menyebabkan nilai f(x) sama dengan nol. Dengan kata lain akar persamaan f(x) adalah titik potong antara kurva f(x) dan sumbu X.

Gnuplot merupakan perangkat lunak pembuatan grafik berbasis command-line. Gnuplot dalam praktikum Metode Numerik nantinya digunakan untuk melakukan pengecekan apakah program yang kita buat sudah menghasilkan keluaran sesuai dengan yang diharapkan.

Berikut ini adalah beberapa fungsi yang didukung dalam gnuplot:

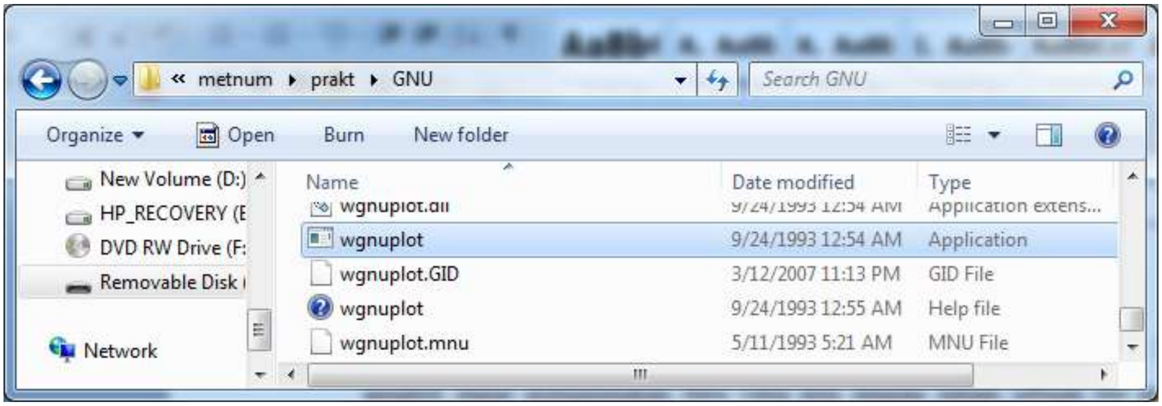
Function	Hasil
-----------	--
abs(x)	nilai mutlak dari x, |x|
acos(x)	arc-cosine dari x
asin(x)	arc-sine dari x

atan(x)	arc-tangent dari x
cos(x)	cosine dari x, x dalam radians.
cosh(x)	hyperbolic cosine dari x, x dalam radians
erf(x)	error function dari x
exp(x)	fungsi exponential dari x, basis e
inverf(x)	fungsi inverse error dari x
invnorm(x)	inverse distribusi normal dari x
log(x)	log dari x, basis e
log10(x)	log dari x, basis 10
norm(x)	fungsi distribusi normal Gaussian
rand(x)	pseudo-random number generator
sgn(x)	1 jika x > 0, -1 jika x < 0, 0 jika x=0
sin(x)	sin dari x, x dalam radians
sinh(x)	hyperbolic sine dari x, x dalam radians
sqrt(x)	akar dari x
tan(x)	tangent dari x, x dalam radians
tanh(x)	hyperbolic tangent dari x, x dalam radians

PLOTING

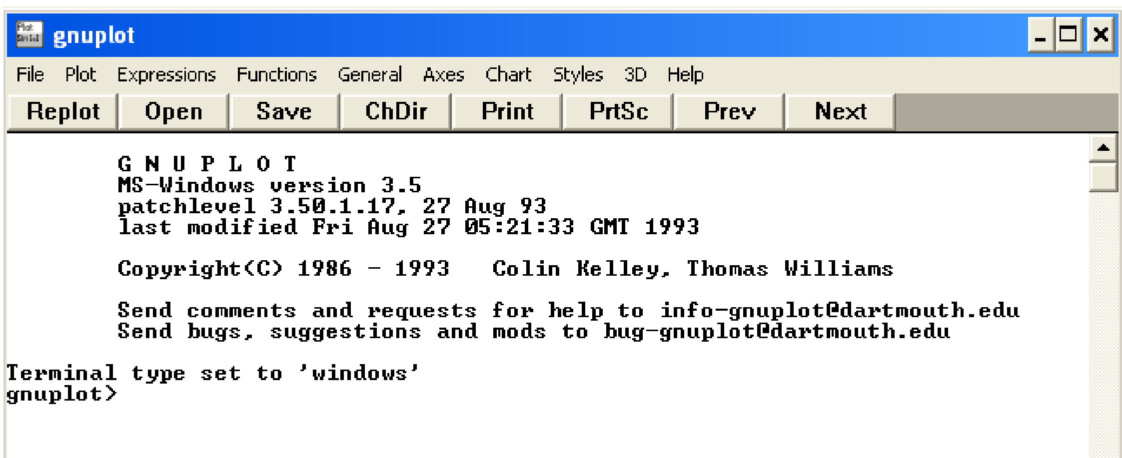
Ploting adalah proses untuk mendapatkan grafik dalam gnuplot. Ploting dalam gnuplot dapat menggunakan data yang kita simpan dalam sebuah file ataupun dengan menggunakan fungsi-fungsi yang sudah disediakan oleh gnuplot seperti di atas. Misalkan kita akan melihat grafik dari fungsi $F(x)=e^{-x} - x$, maka ikuti langkah berikut ini:

1. Jalankan program wgnuplot yang ada pada direktori GNU setelah dilakukan installing



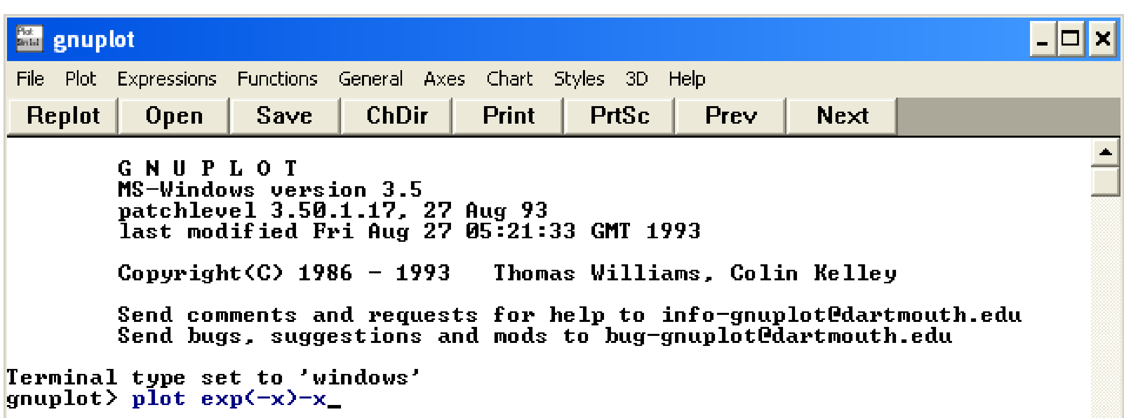
Gambar 2.1 direktori GNU

2. Sehingga muncul tampilan sebagai berikut



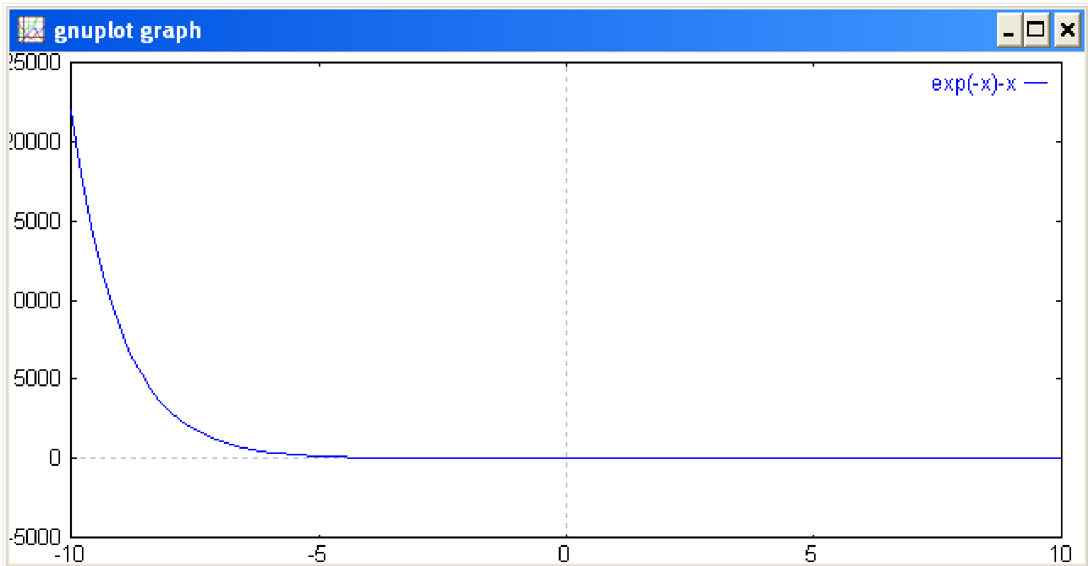
Gambar 2.2 Menu GNUPlot

3. Ketikkan perintah **plot exp(-x)-x** di command prompt-nya, kemudian enter



Gambar 2.3 Menggambar fungsi exponensial

4. Sehingga didapatkan keluaran grafik sebagai berikut

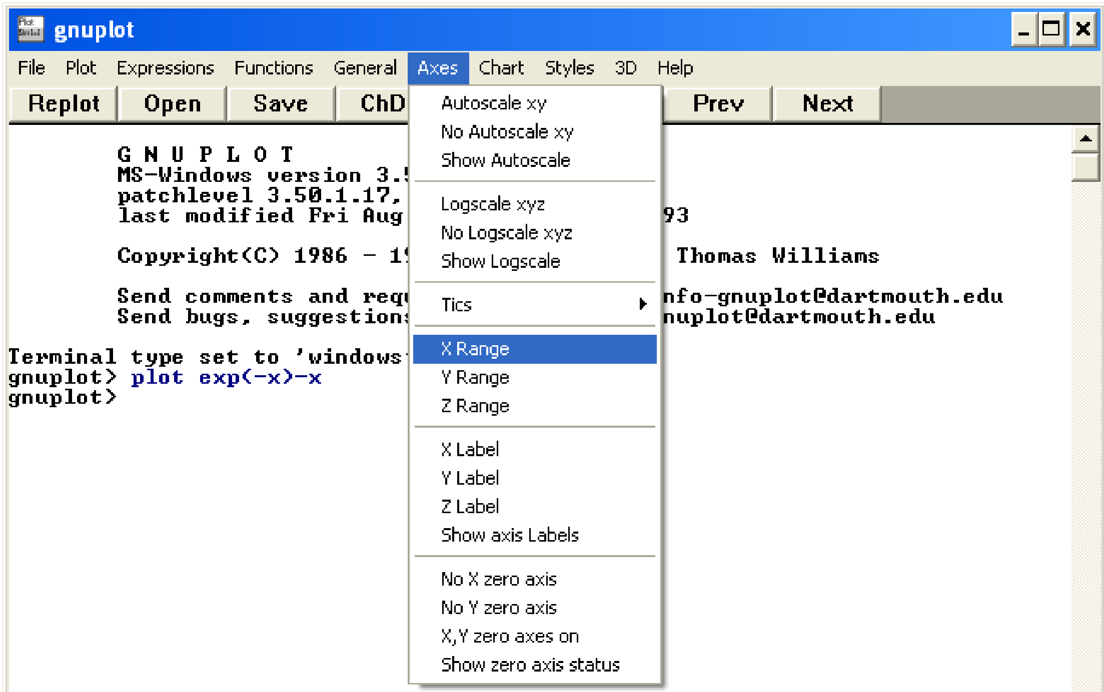


Gambar 2.4 grafik Exponensial

MENENTUKAN RANGE

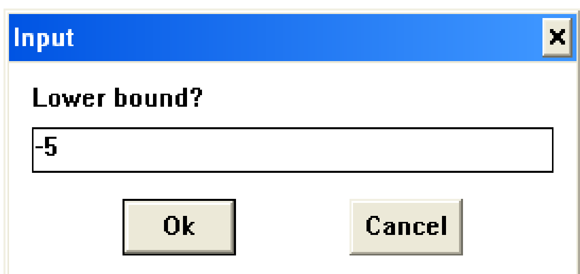
Jika fungsi $F(x)=e^{-x} - x$ di atas kita selesaikan sebagai persamaan linier, nampak bahwa perpotongan fungsi dengan sumbu X tidak jelas pada nilai x berapa. Sehingga dibutuhkan proses penentuan range untuk nilai x dengan jangkauan yang lebih kecil. Untuk proses penentuan range ini, maka ikutilah langkah berikut ini :

1. Lakukan perkiraan pada x berapa terjadi perpotongan dengan y=0. Misal dari fungsi di atas sekitar antara -5 s/d 5.
2. Pilih Axes pada menu, pilih X Range sehingga Nampak seperti gambar berikut:



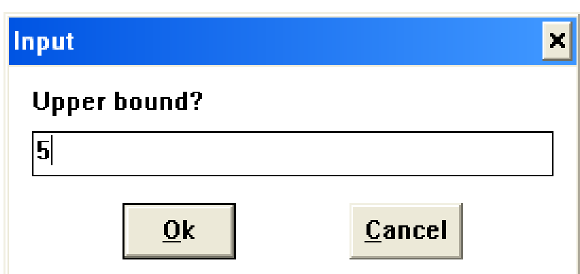
Gambar 2.5 Xrange

3. Isikan -5 pada lower bound



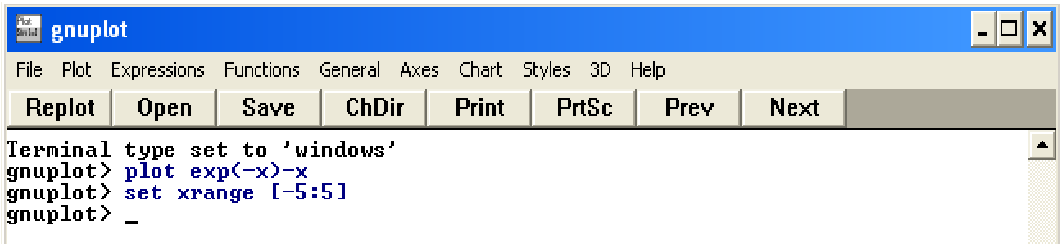
Gambar 2.6 lower bound

4. Isikan 5 pada upper bound



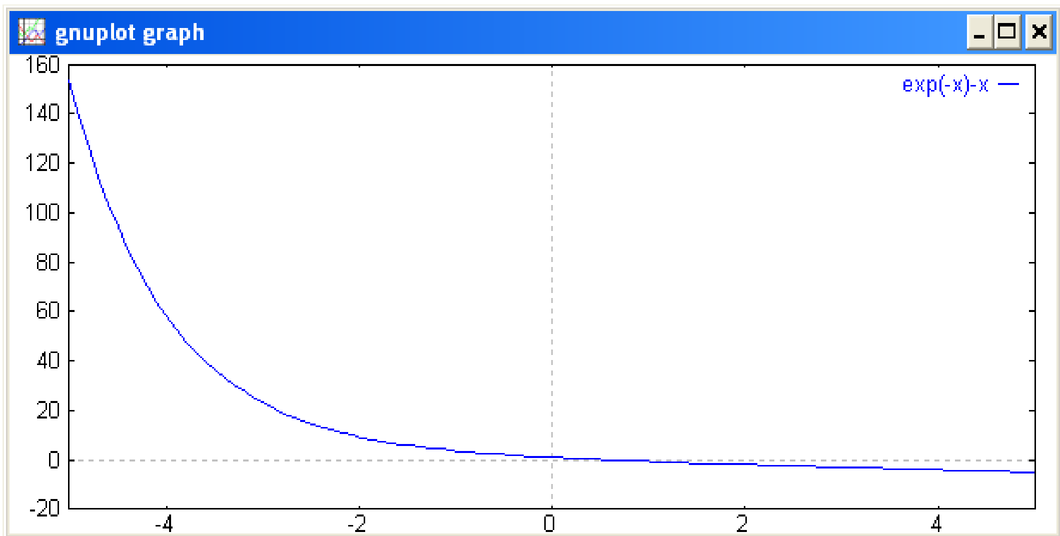
Gambar 2.7 upper bound

5. Sehingga pada command line otomatis terdapat perintah



Gambar 2.8 Gambaran batas pada command line

6. Lakukan plotting lagi dengan mengetikkan plot exp(-x)-x sehingga didapatkan keluaran grafik sebagai berikut:

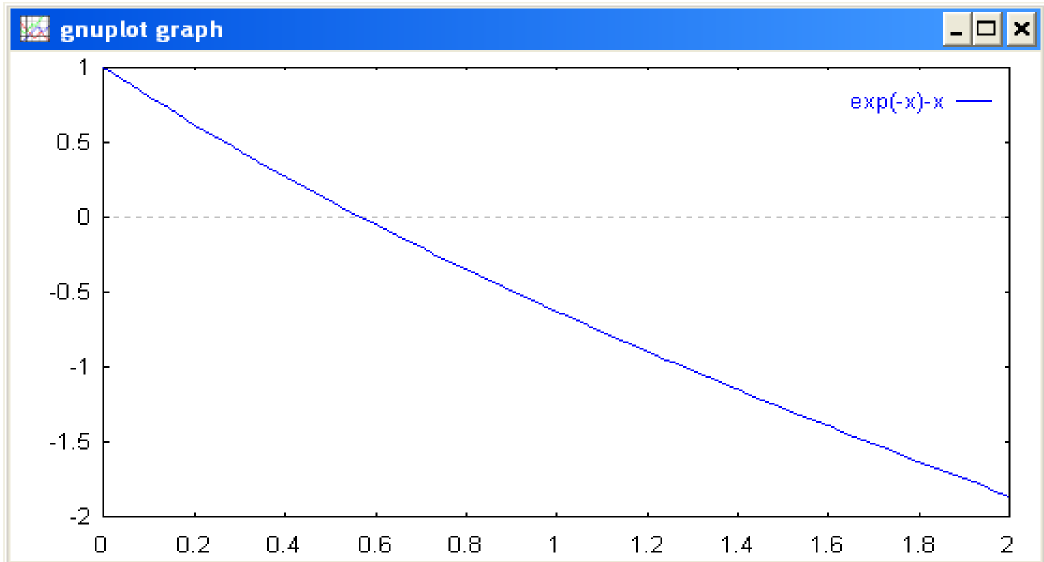


Gambar 2.9 grafik exp(-x)-x

SCALING

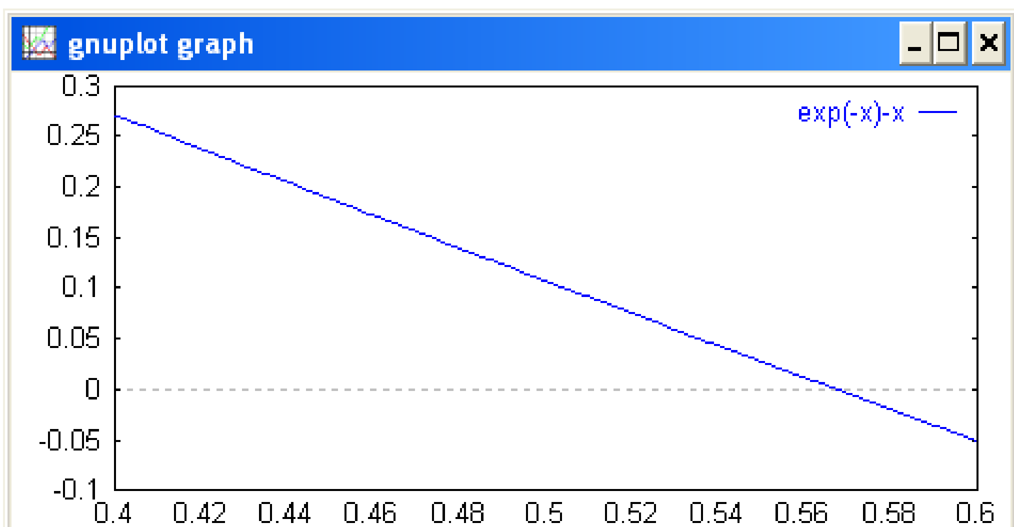
Scaling adalah proses penskalaan untuk mendapatkan nilai x yang akurat pada persamaan linier sehingga didapatkan jawaban nilai x pada perpotongan sumbu X. Untuk mendapatkan nilai x yang akurat adalah dengan mencoba skala yang lebih kecil lagi sampai didapatkan nilai yang akurat.

1. Nampak dari hasil plotting di atas bahwa perpotongan dengan sumbu X terjadi pada nilai x antara 0 sampai dengan 2. Untuk mendapatkan nilai akurat skala dapat diperkecil dengan mengulang perintah set xrange [0:2] pada command prompt dan melakukan ploting lagi.



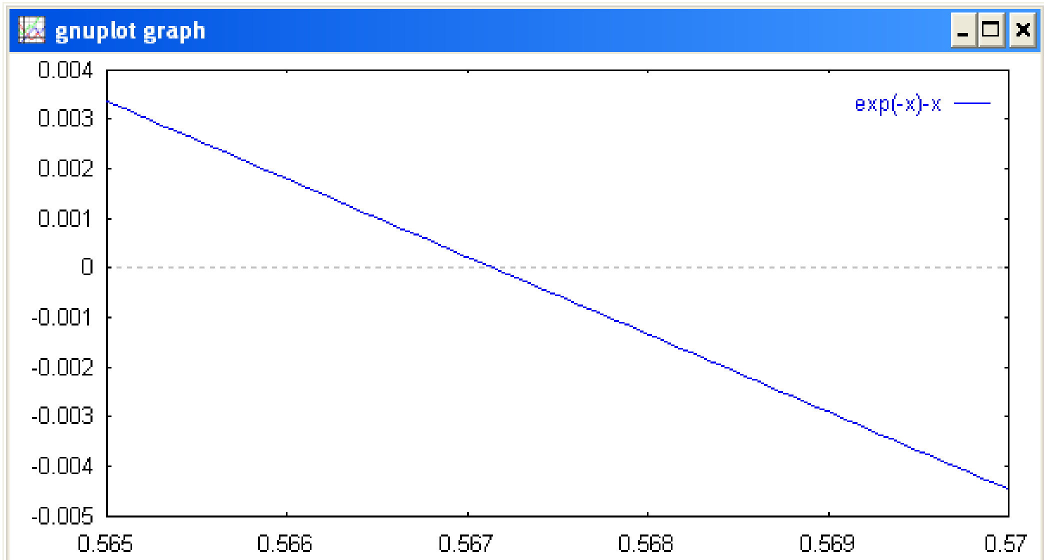
Gambar 2.10 grafik exp(-x)-x pada xrange [0:2]

2. Jika masih belum jelas jangkauannya bisa diperkecil pada x antara 0.4 sampai dengan 0.6 set xrange [0.4:0.6] pada command prompt. Sehingga didapatkan keluaran sebagai berikut:



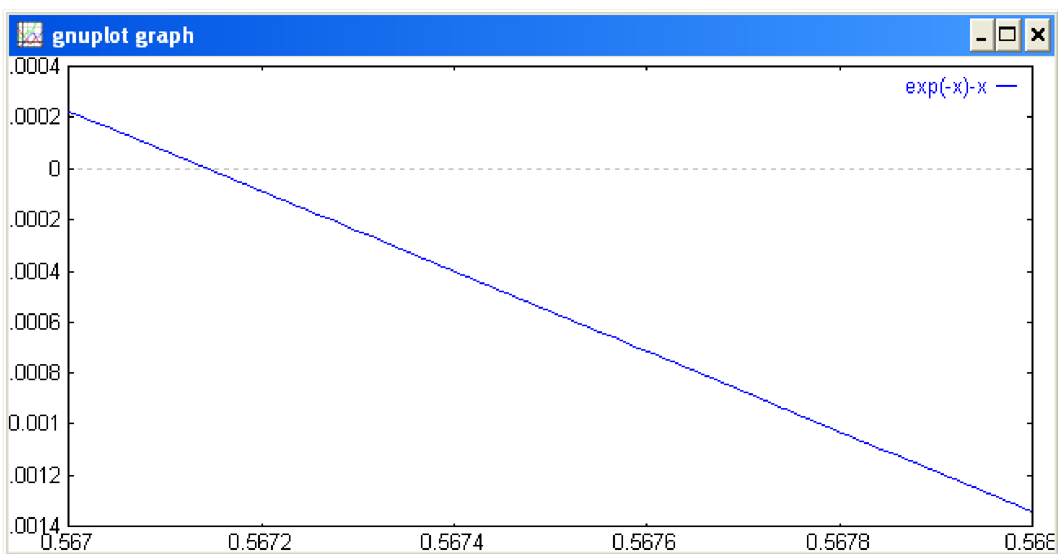
Gambar 2.11 grafik exp(-x)-x pada xrange [0.4:0.6]

3. Dan supaya lebih jelasnya bisa diperkecil pada x antara 0.56 sampai dengan 0.58 set xrange [0.565:0.57] pada command prompt.



Gambar 2.12 grafik exp(-x)-x pada xrange [0.565:0.57]

4. Dan supaya lebih jelasnya bisa diperkecil pada x antara 0.567 sampai dengan 0.568 set xrange [0.567:0.568] pada command prompt. Sehingga pada x=0.5671 merupakan solusi untuk persamaan linier $F(x)=e^{-x} - x$

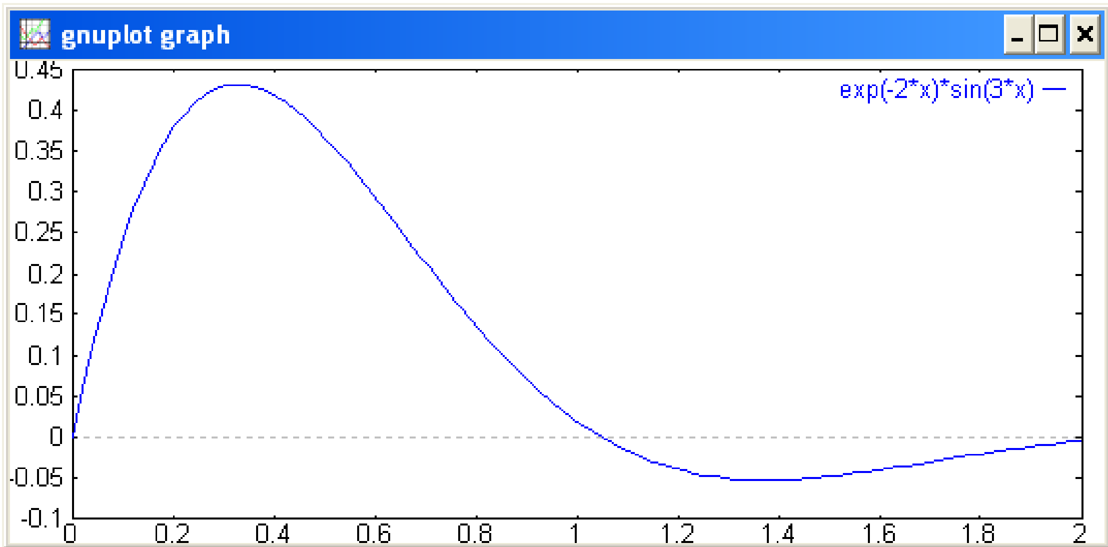


Gambar 2.12 grafik exp(-x)-x pada xrange [0.567:0.568]

MENELUSURI TITIK PUNCAK

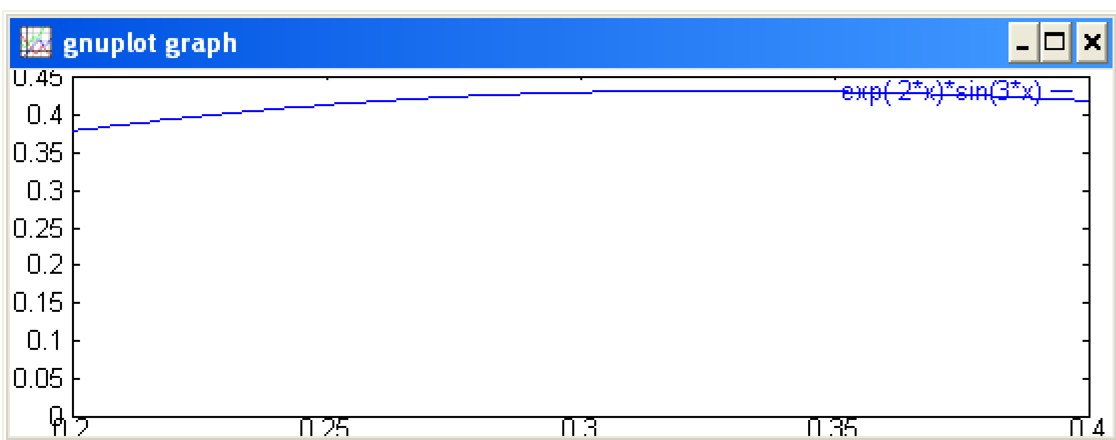
Untuk beberapa fungsi, kita bisa melakukan penelusuran titik puncak. Penelusuran titik puncak ini adalah dengan melakukan penentuan range dan scaling pada sumbu Y. Sebagai contoh jika ada fungsi $F(x)=e^{-2x}.sin(3x)$maka penelusuran titik puncaknya adalah sebagai berikut:

1. Lakukan ploting untuk fungsi $F(x)=e^{-2x}.sin(3x)$ pada x nilai 0 s/d 2



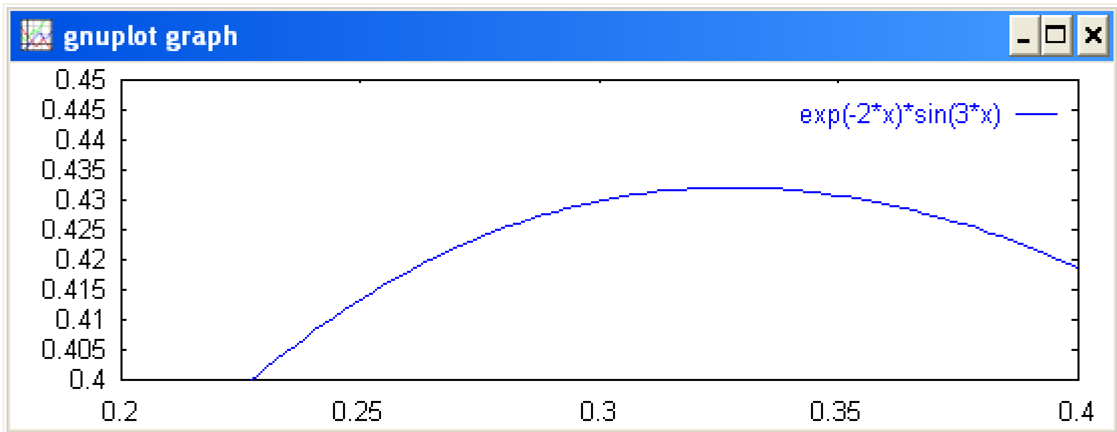
Gambar 2.13 grafik $F(x)=e^{-2x}.sin(3x)$ pada x nilai 0 s/d 2

2. Lakukan scaling pada sumbu X dengan nilai mulai dari 0.2 s/d 0.4 dengan perintah set xrange[0.2:0.4], sehingga didapatkan grafik berikut ini:



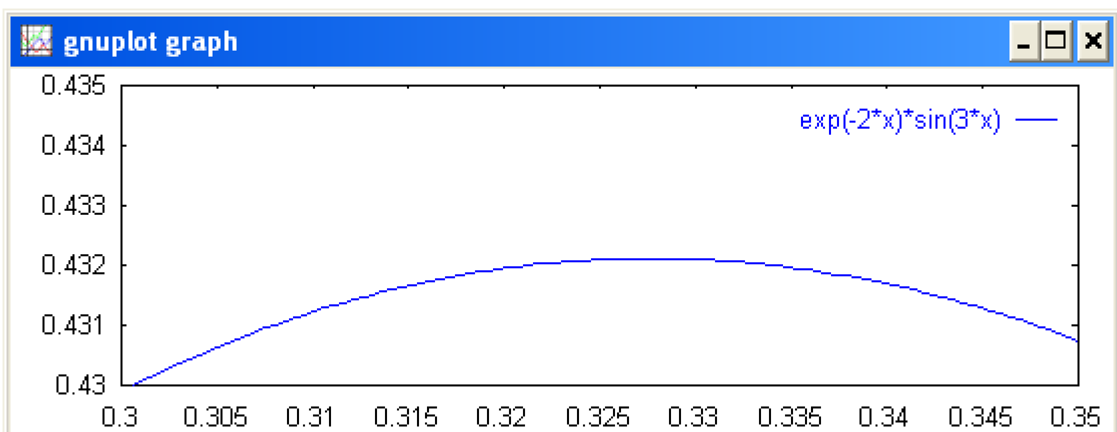
Gambar 2.14 grafik $F(x)=e^{-2x}.sin(3x)$ pada x nilai 0.2 s/d 0.4

3. Lakukan scaling pada sumbu Y dengan nilai mulai dari 0.4 s/d 0.45 dengan perintah set yrange[0.4:0.45], sehingga didapatkan grafik berikut ini



Gambar 2.15 grafik $F(x)=e^{-2x}.sin(3x)$ pada x nilai 0.4 s/d 0.45

4. Lakukan lagi scaling pada sumbu Y dengan nilai mulai dari 0.43 s/d 0.435 dengan perintah set yrange[0.4:0.45] dan scaling pada sumbu X pada nilai 0.3 s/d 0.35, sehingga didapatkan grafik berikut ini



Gambar 2.16 grafik $F(x)=e^{-2x}.sin(3x)$ pada x nilai 0.43 s/d 0.435

5. Didapatkan titik puncak pada nilai x=0.327 dan y=0.432.

C. TUGAS PENDAHULUAN

Instal GNU Plot pada computer saudara , dan bacalah Menu Help.

D. PERCOBAAN

1. Lakukan percobaan untuk langkah-langkah dalam ploting, penentuan range dan scaling yang sudah diuraikan di atas.
2. Lakukan ploting, penentuan range dan scaling untuk fungsi $F(x)=e^{-x} + x$.
3. Lakukan ploting, penentuan range dan scaling untuk fungsi $F(x)=x*e^{-x}+cos(2*x)$
4. Lakukan ploting, penentuan range dan scaling untuk fungsi $F(x)= 2*x^3-exp(-x)$
5. Lakukan ploting, penentuan range dan scaling untuk fungsi *F(x)*= (sqrt(2-x*x)-sqrt(x*x-1)
6. Lakukan ploting, penentuan range, scaling dan penelusuran titik puncak untuk fungsi $F(x)=e^{-2x}.cos(3x)$

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas , sertakan hasil nilai x jika semua fungsi di atas diselesaikan dengan persamaan linier.

PRAKTIKUM 3

PENYELESEIAN PERASAMAAN NONLINIER PART 2

A. TUJUAN PEMBELAJARAN

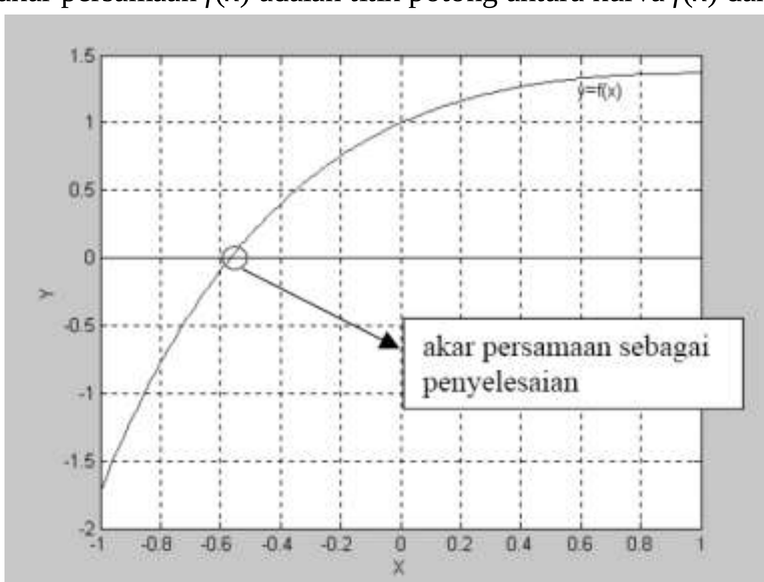
Memahami dan mampu menerapkan :

- Penentuan Range
- Tabeling
- Penggunaan Metode
- Metode Biseksi
- Metode Regula Falsi
- Metode Iterasi Sederhana
- Metode Newton Raphson
- Metode Secant

B. DASAR TEORI

1.1. TEORI PERSAMAAN NON LINEAR

Penyelesaian persamaan non linier adalah penentuan akar-akar persamaan non linier. Akar sebuah persamaan $f(x)$ =0 adalah nilai-nilai x yang menyebabkan nilai f(x)=0. Dengan kata lain akar persamaan $f(x)$ adalah titik potong antara kurva $f(x)$ dan sumbu X.



Gambar 3.1.Penyelesaian Persamaan Non Linear

Penyelesaian persamaan linier biasa yang kita temui adalah $mx + c = 0$ dimana *m* dan *c* adalah konstanta (angka yang berada di depan sebuah variabel), sehingga dapat dihitung dengan:

$$mx + c = 0 \rightarrow x = -\frac{c}{m}$$

Penyelesaiaan persamaan kuadrat $ax^2 + bx + c = 0$ dapat dihitung dengan menggunakan rumus ABC.

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

Beberapa persamaan polinomial yang sederhana dapat diselesaikan theorema sisa. Sehingga tidak memerlukan metode numerik dalam menyelesaikannya, karena metode analitik dapat dilakukan.Tetapi bagaimana menyelesaikan persamaan yang mengandung unsur bilangan natural, seperti :

$$x - e^{-x} = 0$$

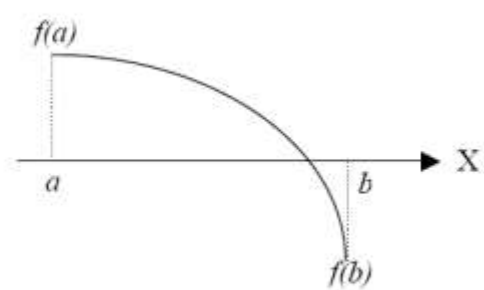
Tampaknya sederhana, tetapi untuk menyelesaikan persamaan non linear merupakan metode pencarian akar secara berulang-ulang.

1.2. PENYELESAIAN MENCARI NILAI AKAR DARI PERSAMAAN NON LINEAR

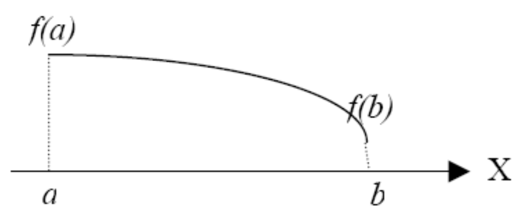
1.2.1. METODE TERTUTUP

A. METODE TABEL

Suatu range $x = [a, b]$ mempunyai akar bila *f(a)* dan *f(b)* berlawanan tanda atau memenuhi *f(a).f(b)<0*. Lihat gambar grafik di bawah ini :



Karena *f*(*a*).*f*(*b*)<0 maka pada range *x*=[a,b] terdapat akar.



Karena *f*(*a*).*f*(*b*)>0 maka pada range *x*=[a,b] tidak ada akar.

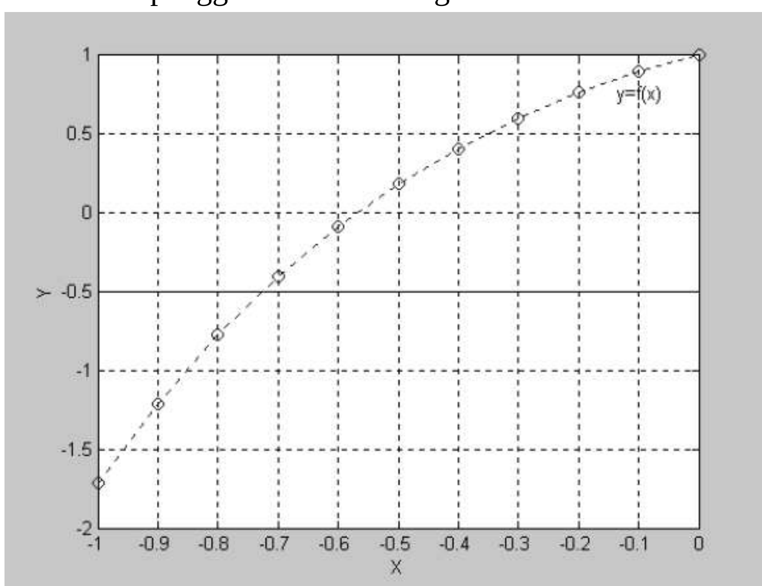
Gambar 3.2: Grafik Penetuan akar Persamaan

Secara sederhana, untuk menyelesaikan persamaan non linier dapat dilakukan dengan menggunakan metode table 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 sebuah tabel.

Dari tabel ini, bila ditemukan $f(x_k) = 0$ atau mendekati nol maka dapat dikatakan bahwa x_k adalah penyelesaian persamaan $f(x_k) = 0$. Bila tidak ada $f(x_k)$ yang sama dengan nol, maka dicari nilai $f(x_k)$ dan $f(x_{k+1})$ yang berlawanan tanda, bila tidak ditemukan maka dikatakan tidak mempunyai akar untuk x = [*a*,*b*], dan bila ditemukan maka ada 2 pendapat untuk menentukan akar persamaan :

- Bila $| f(x_i) | < | f(x_{i+1}) |$ maka x_k adalah penyelesaian
- Bila tidak x_{k+1} adalah penyelesaian atau dapat dikatakan penyelesaian berada di antara x_k dan x_{k+1}.

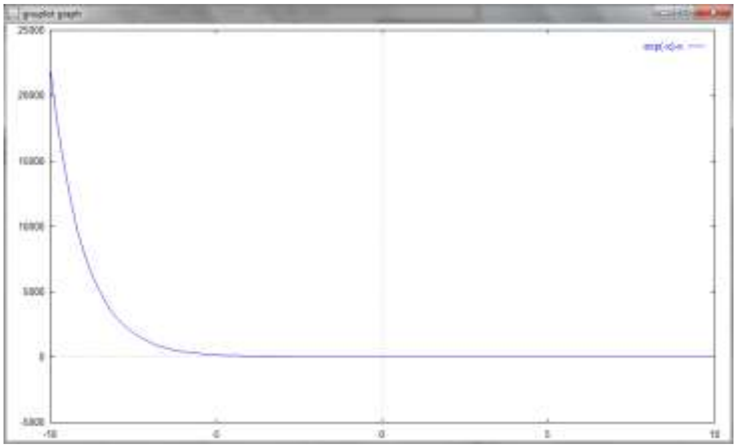
Berikut ini adalah penggambaran secara grafis dari metode tabel:



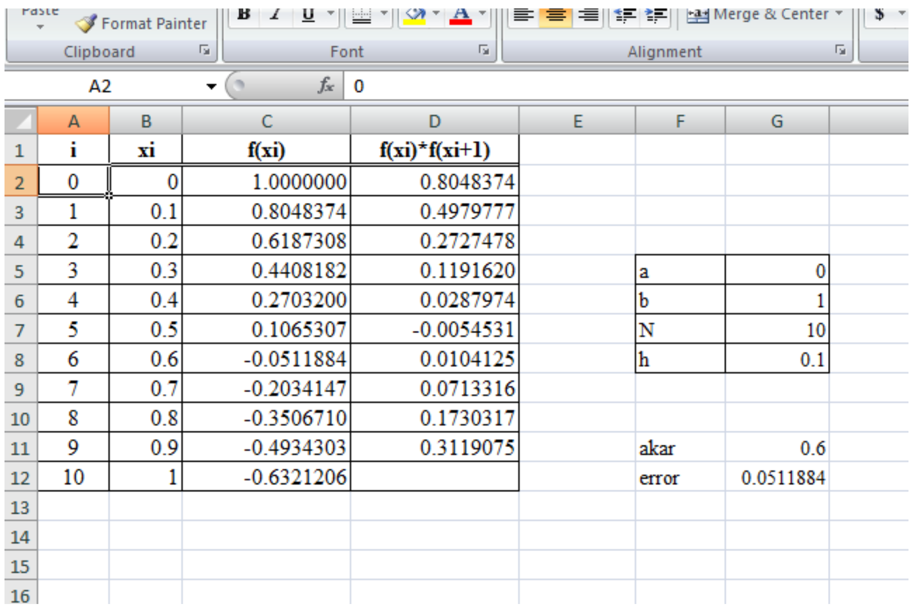
Gambar 3.3: Metode Tabel secara Grafis

Gambar di atas menjelaskan bahwa penyelesaian diperoleh dengan membagi *x* = [*a*,*b*] sebanyak-banyaknya hingga diperoleh suatu garis yang melalui akar persamaan dan nilai *x* dari garis tersebut adalah penyelesaian dari persamaan $f(x) =$ 0.

Sebagai contoh, cari akar dari persamaan $f(x_i) = e^{-x} - x$. Gunakan Microsoft Excel untuk membuat sketsa program dari metode tabel.



Gambar 3.4: Grafik fungsi $f(x) = e^{-x} - x$ *dengan GNU Plot*



	A	B	C	D	E	F	G
1	i	xi	f(xi)	f(xi)*f(xi+1)			
2	0	0	1.0000000	0.8048374			
3	1	0.1	0.8048374	0.4979777			
4	2	0.2	0.6187308	0.2727478			
5	3	0.3	0.4408182	0.1191620		a	0
6	4	0.4	0.2703200	0.0287974		b	1
7	5	0.5	0.1065307	-0.0054531		N	10
8	6	0.6	-0.0511884	0.0104125		h	0.1
9	7	0.7	-0.2034147	0.0713316			
10	8	0.8	-0.3506710	0.1730317			
11	9	0.9	-0.4934303	0.3119075		akar	0.6
12	10	1	-0.6321206			error	0.0511884

Gambar 3.5: Gambar penggunaan metode tabel dengan Ms. Excel

Algoritma Metode Tabel :

(1) Defisikan fungsi f(x)

(2) Tentukan range untuk x yang berupa batas bawah x_{bawah}dan batas atas x_{atas}.

(3) Tentukan jumlah pembagian N

(4) Hitung step pembagi h

$$H = \frac{x_{atas} - x_{bawah}}{N}$$

(5) Untuk i = 0 s/d N, hitung

$x_i = x_{bawah} + i.h$

$y_i = f(x_i)$

(6) Untuk I = 0 s/d N dicari k dimana

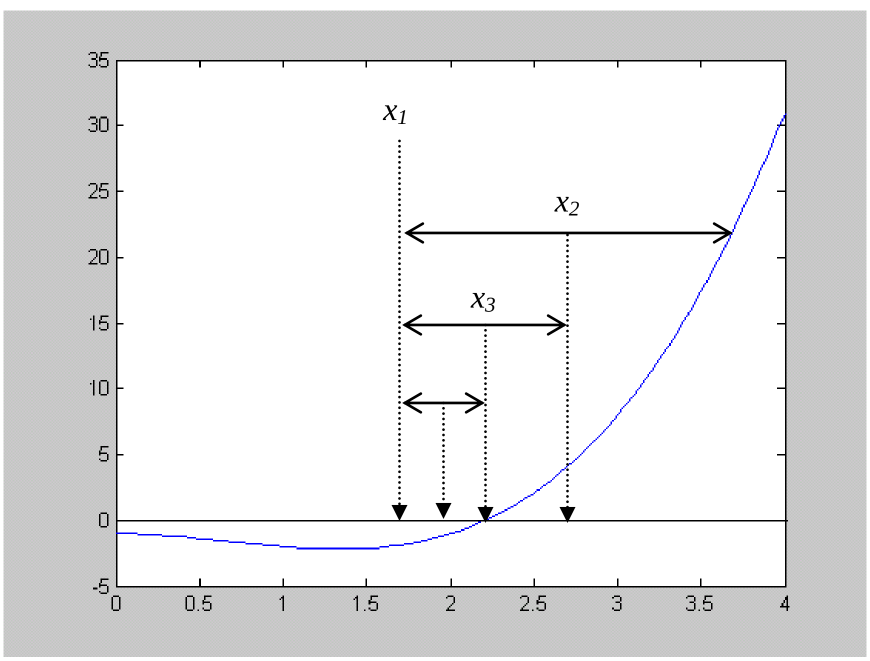
*. Bila $f(x_k) = 0$ maka x_k adalah penyelesaian

*. Bila $f(x_k).f(x_{k+1}) < 0$ maka :

- Bila $|f(x_k)| <|f(x_{k+1})$ maka x_k adalah penyelesaian
- Bila tidak x_{k+1}adalah penyelesaian atau dapat dikatakan penyelesaian berada di antara x_k dan x_{k+1}.

B. Metode Biseksi

Ide awal metode ini adalah metode table, dimana area dibagi menjadi N bagian.Hanya saja metode biseksi ini membagi range menjadi 2 bagian, dari dua bagian ini dipilih bagian mana yang mengandung dan bagian yang tidak mengandung akar dibuang.Hal ini dilakukan berulang-ulang hingga diperoleh akar persamaan.



Gambar 3.6. Metode Biseksi

Untuk menggunakan metode biseksi, terlebih dahulu ditentukan batas bawah (*a*) dan batas atas (*b*).Kemudian dihitung nilai tengah :

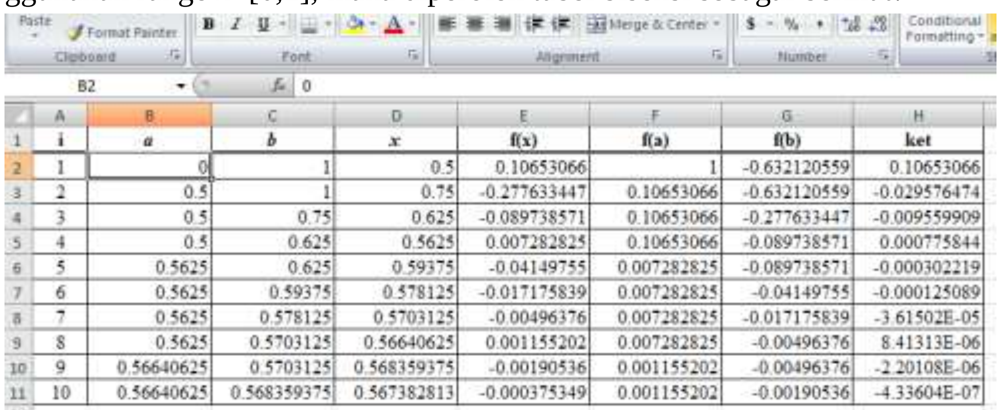
$$x = \frac{a+b}{2}$$

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

$f(a) \cdot f(b) < 0$

Setelah diketahui dibagian mana terdapat akar, maka batas bawah dan batas atas di perbaharui sesuai dengan range dari bagian yang mempunyai akar.

Sebagai contoh, untuk menyelesaikan persamaan $e^{-x} - x = 0$, dengan menggunakan range *x*=[0,1], maka diperoleh tabel biseksi sebagai berikut:



i	a	b	x	f(x)	f(a)	f(b)	ket
1	0	1	0.5	0.10653066	1	-0.632120559	0.10653066
2	0.5	1	0.75	-0.277633447	0.10653066	-0.632120559	-0.029576474
3	0.5	0.75	0.625	-0.089738571	0.10653066	-0.277633447	-0.009559909
4	0.5	0.625	0.5625	0.007282825	0.10653066	-0.089738571	0.000775844
5	0.5625	0.625	0.59375	-0.04149755	0.007282825	-0.089738571	-0.000302219
6	0.5625	0.59375	0.578125	-0.017175839	0.007282825	-0.04149755	-0.000125089
7	0.5625	0.578125	0.5703125	-0.00496376	0.007282825	-0.017175839	-3.61502E-05
8	0.5625	0.5703125	0.56640625	0.001155202	0.007282825	-0.00496376	8.41313E-06
9	0.56640625	0.5703125	0.568359375	-0.00190536	0.001155202	-0.00496376	-2.20108E-06
10	0.56640625	0.568359375	0.567382813	-0.000375349	0.001155202	-0.00190536	-4.33604E-07

Gambar 3.7: Gambar penggunaan metode Biseksi dengan Ms. Excel

Algoritma Metode Biseksi:

(1) Definisikan fungsi *f(x)* yang akan dicari akarnya

(2) Tentukan nilai *a* dan *b*

(3) Tentukan torelansi e dan iterasi maksimum *N*

(4) Hitung *f(a)* dan *f(b)*

(5) Jika *f(a).f(b)>0* maka proses dihentikan karena tidak ada akar, bila tidak dilanjutkan

(6) Hitung xr = $\dfrac{a+b}{2}$

(7) Hitung *f(xr)*

(8) Bila *f(xr).f(a)<0* maka *b=xr* dan *f(b)=f(xr)*, bila tidak *a=xr* dan *f(a)=f(xr)*

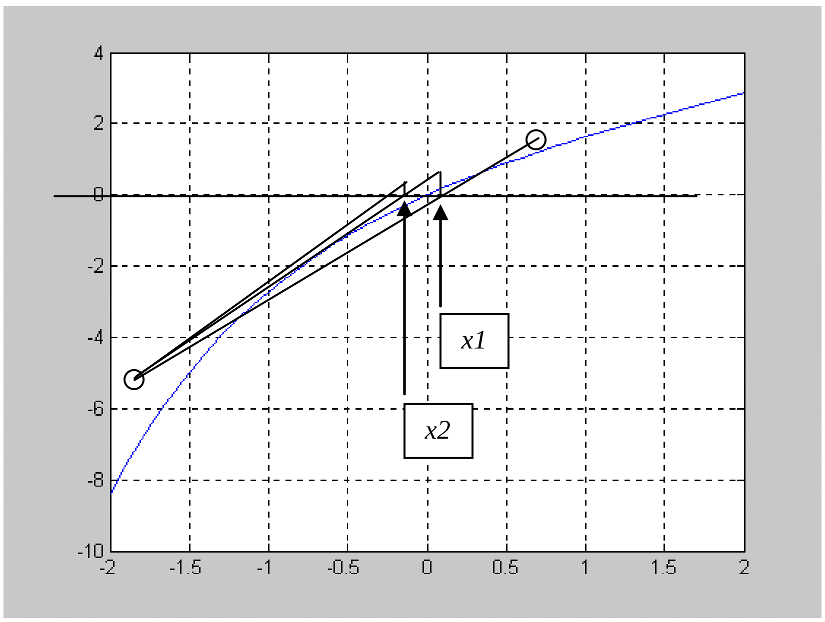
(9) Jika |b-a|<e atau iterasi>iterasi maksimum maka proses dihentikan dan didapatkan akar = xr, dan bila tidak, ulangi langkah 6.

C. Metode Regula Falsi

Metode regula falsi adalah metode pencarian akar persamaan dengan memanfaatkan kemiringan dan selisih tinggi dari dua titik batas range. Seperti halnya metode biseksi, metode ini bekerja secara iterasi dengan melakukan update range.Titik pendekatan yang digunakan oleh metode regula-falsi adalah :

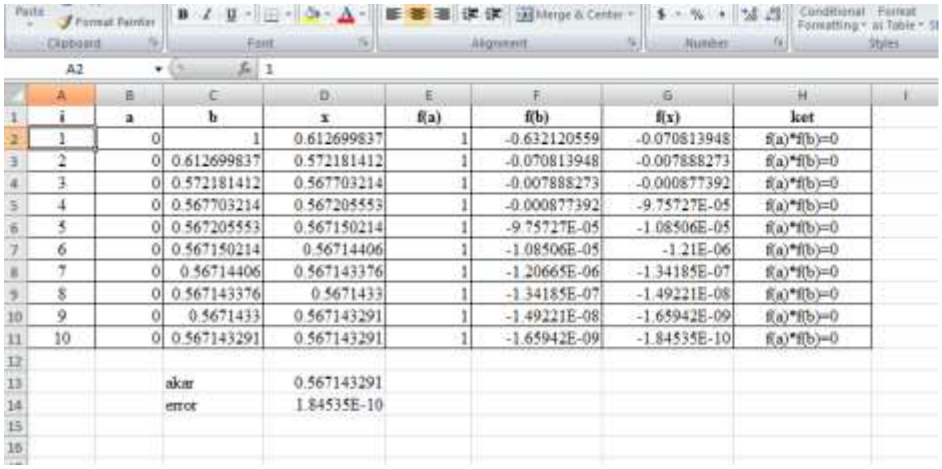
$$X = \frac{f(b).a - f(a).b}{f(b) - f(a)}$$

Dengan kata lain titik pendekatan x adalah nilai rata-rata range berdasarkan F(x).Metode regula falsi secara grafis digambarkan sebagai berikut :



Gambar 3.8. Metode Regula Falsi

Contoh tabel sketsa program dengan fungsi $f(x) = e^{-x} - x$:



i	a	b	x	f(a)	f(b)	f(x)	ket
1	0	1	0.612699837	1	-0.632120559	-0.070813948	f(a)*f(b)=0
2	0	0.612699837	0.572181412	1	-0.070813948	-0.007888273	f(a)*f(b)=0
3	0	0.572181412	0.567703214	1	-0.007888273	-0.000877392	f(a)*f(b)=0
4	0	0.567703214	0.567205553	1	-0.000877392	-9.75727E-05	f(a)*f(b)=0
5	0	0.567205553	0.567150214	1	-9.75727E-05	-1.08506E-05	f(a)*f(b)=0
6	0	0.567150214	0.56714406	1	-1.08506E-05	-1.21E-06	f(a)*f(b)=0
7	0	0.56714406	0.567143376	1	-1.20665E-06	-1.34185E-07	f(a)*f(b)=0
8	0	0.567143376	0.5671433	1	-1.34185E-07	-1.49221E-08	f(a)*f(b)=0
9	0	0.5671433	0.567143291	1	-1.49221E-08	-1.65942E-09	f(a)*f(b)=0
10	0	0.567143291	0.567143291	1	-1.65942E-09	-1.84535E-10	f(a)*f(b)=0
		akar	0.567143291				
		error	1.84535E-10				

Gambar 3.9: Gambar penggunaan metode Regula Falsi dengan Ms. Excel

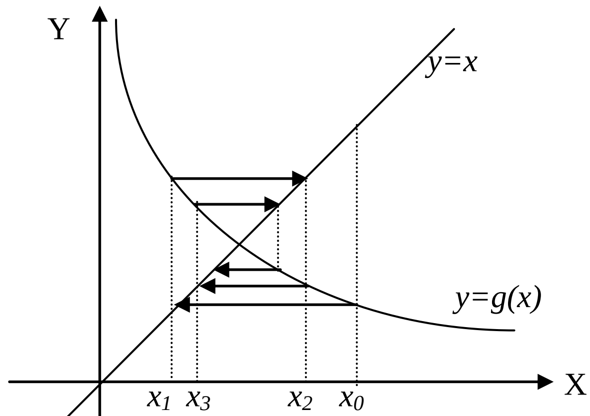
Algoritma Metode Regula Falsi:

1. Definisikan fungsi f(x)
2. Tentukan batas bawah (a) dan batas atas (b)
3. Tentukan toleransi error (e) dan iterasi maksimum (N)
4. Hitung Fa = f(a) dan Fb = f(b)
5. Untuk iterasi I = 1 s/d n atau error > e
   - xr = $\dfrac{F(b).a - F(a).b}{F(b) - F(a)}$
   - Hitung Fx = f(x)
   - Hitung error = |Fx|
   - Jika Fx.Fa <0 maka b = xr dan Fb = Fxr jika tidak a = xr dan Fa = Fxr.
6. Akar persamaan adalah xr.

1.2.2. METODE TERBUKA

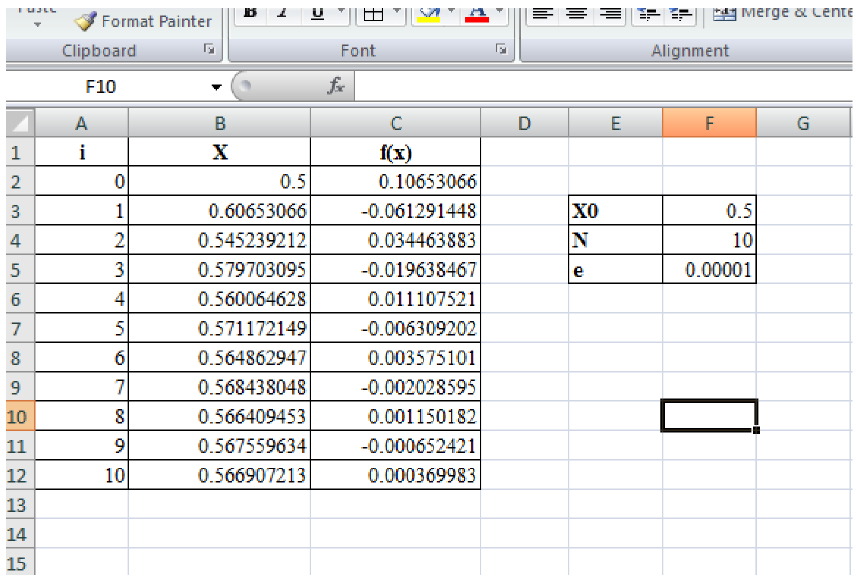
1. Metode Iterasi

Metode iterasi sederhana adalah metode yang memisahkan x dengan sebagian x yang lain sehingga diperoleh : x = g(x). Sebagai contoh untuk menyelesaikan persamaan $x - e^x = 0$ maka persamaan di ubah menjadi : $x = e^x$ atau $g(x) = e^x.g(x)$ inilah yang menjadi dasar iterasi pada metode iterasi sederhana ini.Metode iterasi sederhana secara grafis dapat dijelaskan sebagai berikut :



Gambar 3.10. Metode Iterasi Sederhana

Contoh tabel sketsa program untuk fungsi $f(x) = e^{-x} - x$:



	A	B	C	D	E	F	G
1	**i**	**X**	**f(x)**				
2	0	0.5	0.10653066				
3	1	0.60653066	-0.061291448		**X0**	0.5	
4	2	0.545239212	0.034463883		**N**	10	
5	3	0.579703095	-0.019638467		**e**	0.00001	
6	4	0.560064628	0.011107521				
7	5	0.571172149	-0.006309202				
8	6	0.564862947	0.003575101				
9	7	0.568438048	-0.002028595				
10	8	0.566409453	0.001150182				
11	9	0.567559634	-0.000652421				
12	10	0.566907213	0.000369983				
13							
14							
15							

Gambar 3.11: Gambar penggunaan metode Iterasi Sederhana dengan Ms. Excel

Algoritma Metode Iterasi Sederhana:

1. Definisikan F(x) dan g(x)
2. Tentukan toleransi error (e) dan iterasi maksimum (n)
3. Tentukan pendekatan awal $x[0]$
4. Untuk iterasi = 1 s/d n atau $F(x[iterasi]) \geq e$

   $X_i = g(x_{i-1})$

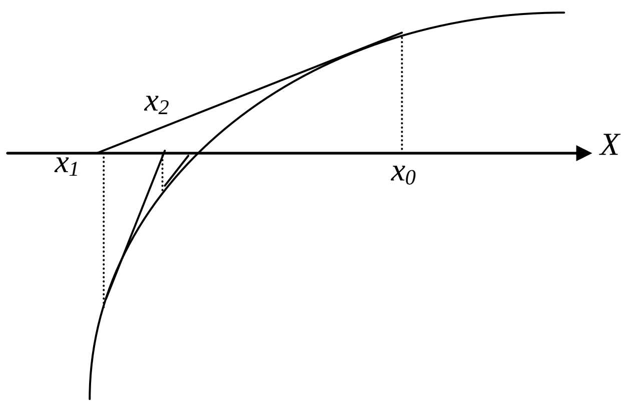
   Hitung $F(x_i)$
5. Akar adalah x terakhir yang diperoleh.

2. Metode newton raphson

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

$$X_{n+1} = x_n + \frac{F(x_n)}{F^1(x_n)}$$

Metode newton raphson dapat digambarkan sebagai berikut :



Gambar.3..12 Metode Newton Raphson.

Contoh tabel sketsa program untuk fungsi $f(x) = e^{-x} - x$:

Clipboard | Font | Alignment | Number

B6 | =B5-(C5/D5)

	A	B	C	D
1	i	xo	f(x)	f'(x)
2	0	0	1.000000000	-2
3	1	0.5	0.106530660	-1.60653066
4	2	0.566311003	0.001304510	-1.567615513
5	3	0.567143165	0.000000196	-1.567143362
6	4	0.56714329	0.000000000	-1.56714329
7	5	0.56714329	0.000000000	-1.56714329
8	6	0.56714329	0.000000000	-1.56714329
9	7	0.56714329	0.000000000	-1.56714329
10	8	0.56714329	0.000000000	-1.56714329
11	9	0.56714329	0.000000000	-1.56714329
12	10	0.56714329	0.000000000	-1.56714329
13				

Gambar 3.13: Gambar penggunaan metode Iterasi Sederhana dengan Ms. Excel

Algoritma Metode Newton Raphson:

1. Definisikan fungsi f(x) dan $f^1(x)$
2. Tentukan toleransi error (e) dan iterasi maksimum (n)
3. Tentukan nilai pendekatan awal x_0
4. Hitung $f(x_0)$ dan $f^1(x_0)$
5. Untuk iterasi I = 1 s/d n atau $|f(x_i)| \geq e$

   $x_{i+1} = x_i - \dfrac{f(x_i)}{f^1(x_i)}$

   Hitung $f(x_i)$ dan $f^1(x_i)$
6. Akar persamaan adalah nilai x_i yang terakhir diperoleh.

3. Metode secant

Metode secant merupakan perbaikan dari metode regula-falsi dan newton raphson dimana kemiringan dua titik dinyatakan sacara diskrit, dengan mengambil bentuk garis lurus yang melalui satu titik.

$y - y_0 = m(x - x_0) atau$, dimana m diperoleh dari $m_n = \dfrac{F(x_n) - F(x_{n-1})}{x_n - x_{n-1}}$

Bila y = F(x), n_y dan x_n diketahui maka titik ke n+1 adalah :

$y_{n+1} - y_n = m_n(x_{n+1} - x_n)$

Bila titik x_{n+1} dianggap akar persamaan maka :

$Y_{n+1}=0$ sehingga diperoleh : $-y_n = m_n(x^{n+1} - x_n)$

$$\frac{m_n x_n - y_n}{m_n} = x_{n+1}$$

atau : $x_{n+1} = x_n - y_n \cdot \dfrac{1}{m_n}$

$$x_{n+1} = x_n - y_n \frac{x_n - x_{n+1}}{y_n - y_{n+1}}$$

Persamaan ini yang menjadi dasar pada proses pendekatan dimana nilai pendekatannya adalah :

$$\delta_n = -y_n \frac{x_n - x_{n+1}}{y_n - y_{n+1}}$$

Sehingga untuk menggunakan metode secant ini diperlukan dua titik pendekatan x_0 dan x_1. Kedua titik pendekatan ini diambil pada titik-titik yang dekat agar konvergensinya dapat dijamin.

Algoritma Metode Secant :

1. Definisikan fungsi F(x)
2. Ambil range nilai x = $[a, b]$ dengan jumlah pembagi p
3. Masukkan torelansi error (e) dan masukkan iterasi n
4. Gunakan algoritma tabel diperoleh titik pendekatan awal x_0 dan x1 untuk setiap range yang diperkirakan terdapat akar dari :

   $F(x_k) * F(x_{k+1})<0$ maka $x_0 = x_k$ dan $x_1=x_0+(b-a)/p$. Sebaiknya gunakan metode tabel atau grafis untuk menjamin titik pendakatannya adalah titik pendekatan yang konvergensinya pada akar persamaan yang diharapkan.
5. Hitung $F(x_0)$ dan $F(x_1)$ sebagai y_0 dan y_1
6. Untuk iterasi I = 1 s/d n atau $|F(x_i)| \geq e$

   $$x_{i+1} = x_i - y_i \frac{x_i - x_{i-1}}{y_i - y_{i-1}}$$

   Hitung $y_{i+1} = F(x_{i+1})$
7. Akar persamaan adalah nilai x yang terakhir.

C. TUGAS PENDAHULUAN

Tuliskan dasar-dasar komputasi dari metode terbuka dan tertutup untuk penyeleseian non linier, Yang berisi : dasar teori , algoritma dan flowchart.

D. PERCOBAAN

1. Didefinisikan persoalan dari persamaan non linier dengan fungsi sebagai berikut : $f(x) = 1/(x-2)$
2. Pengamatan awal
   a. Gunakan Gnu Plot untuk mendapatkan kurva fungsi persamaan
   b. Amati kurva fungsi yang memotong sumbu x
   c. Dapatkan dua nilai pendekatan awal diantara nilai x (b) sebagai nilai a (=batas bawah) dan nilai b (=batas atas)

3. Penulisan hasil
    a. Seleseikan permasalahan diatas dengan menggunakan salah satu metode tertutup
    b. Seleseikan permasalahan diatas dengan menggunakan salah satu metode terbuka dengan menggunakan titik awal diantara a dan b.
    c. Akhir iterasi ditentukan sampai dengan 20 iterasi
4. Pengamatan terhadap hasil dengan macam-macam parameter input
    a. Nilai error (e) akar ditentukan = 0.0001 sebagai pembatas iterasi nilai f(x)
    b. Jumlah iterasi maksimum
    c. Bandingkan antara 3a dan 3b terhadap hasil yang diperoleh
    d. Pengubahan nilai awal batas bawah dan batas atas

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas

FORM LAPORAN AKHIR
Nama dan NRP mahasiswa

Judul Percobaan : METODE Tertutup

Algoritma :

Listing program yang sudah benar :

Pengamatan awal

1. Gambar kurva fungsi dengan Gnu Plot
2. Perkiraan nilai x0

Hasil percobaan :

1. Tabel hasil iterasi, xi, f(xi)
1. Pengamatan terhadap parameter
   a. Toleransi error(e) terhadap jumlah iterasi (N)

Toleransi Error (e)	Jumlah Iterasi (N)
0.1	
0.01	
0.001	
0.0001	

   b. Perubahan nilai awal x0 terhadap iterasi (N)

Batas Bawah (a)	Batas Atas (b)	Nilai Error (F(x)=e)

Buatlah kesimpulan dari jawaban 2a dan 2b, kemudian gambarkan grafiknya

FORM LAPORAN AKHIR
Nama dan NRP mahasiswa

Judul Percobaan : METODE Terbuka
Algoritma :

Listing program yang sudah benar :

Pengamatan awal
3. Gambar kurva fungsi dengan Gnu Plot
4. Perkiraan nilai x0

Hasil percobaan :
2. Tabel hasil iterasi, xi, f(xi)
2. Pengamatan terhadap parameter
   a. Toleransi error(e) terhadap jumlah iterasi (N)

Toleransi Error (e)	Jumlah Iterasi (N)
0.1	
0.01	
0.001	
0.0001	

   b. Perubahan nilai awal x0 terhadap iterasi (N)

X0	Iterasi

Buatlah kesimpulan dari jawaban 2a dan 2b, kemudian gambarkan grafiknya

PRAKTIKUM 4

PERSAMAAN LINIER SIMULTAN_1

A. TUJUAN PEMBELAJARAN

Memahami dan mampu menerapkan :

- OBE
- Metode Eliminasi Gauss
- Metode Gauss Jordan

B. DASAR TEORI

Metode Eliminasi Gauss merupakan metode yang dikembangkan dari metode eliminasi, yaitu menghilangkan atau mengurangi jumlah variable sehingga dapat diperoleh nilai dari suatu variable bebas. Cara eliminasi ini sudah banyak dikenal. Untuk menggunakan metode eliminasi Gauss ini, terlebih dahulu bentuk matrik diubah menjadi augmented matrik sebagai berikut :

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

Metode eliminasi gauss, adalah suatu metode dimana bentuk matrik di atas, pada biagan kiri diubah menjadi matrik segitiga atas atau segitiga bawah dengan menggunakan **OBE** (**Operasi Baris Elementer**).

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

Sehingga penyelesaian dapat diperoleh dengan:

$$x_n = \frac{d_n}{c_{nn}}$$

$$x_{n-1} = \frac{1}{c_{n-1,n-1}}\left(-c_{n-1,n}x_n + d_{n-1}\right)$$

....

$$x_2 = \frac{1}{c_{22}}\left(d_2 - c_{23}x_3 - c_{24}x_4 - \dots - c_{2n}x_n\right)$$

$$x_1 = \frac{1}{c_{11}}\left(d_1 - c_{12}x_2 - c_{13}x3 - \dots - c_{1n}x_n\right)$$

Operasi Baris Elementer (OBE) merupakan suatu operasional pengubahan nilai elemen matrik berdasarkan barisnya, tanpa mengubah matriknya. OBE pada baris ke-i+k dengan dasar baris ke i dapat dituliskan dengan :

$$a_{i+k,j} = a_{i+k,j} - c.a_{i,j}$$

dimana c adalah konstanta pengali yang diambil dari perbandingan nilai dari elemen $a_{i,i}$ dan $a_{i+k,i}$

Algoritma Eliminasi Gauss-tanpa pivoting
(asumsi : indeks array selalu dimulai dari 0)

1. Masukkan jml ordo matriks n dan inputkan masing-masing elemen augmented matriks A

2. Untuk kolom ke-i, di mana : 0 <= i < n-1 lakukan OBE (operasi baris elementer) sbb :
   Untuk baris ke-j, di mana: i+1 <= j < n
   - hitung nilai konstanta c : c = A[j][i] / A[i][i]
   - Untuk kolom ke-k, di mana : 0 <= k < n+1 (termasuk kolom yg berisi vektor b) hitunglah :
     A[j][k] = A[j][k] - c * A[i][k]

3. Isikan elemen matriks pada kolom ke-n (kolom terakhir dari augmented matriks) ke dalam matriks vektor b sbb :
   Untuk baris ke-i : 0 <= i < n
   b[i] = A[i][n]

4. Hitung akar x dengan cara melakukan substitusi mundur sbb :
   - x[n-1] = b[n-1]/A[n-1][n-1]
   - Untuk baris ke-k, di mana : n-2 >= k >= 0
     - sigma = 0
     - untuk kolom ke-j, di mana : k+1 <= j < n , hitunglah
       sigma = sigma + A[k][j] * x[j]
     - x[k] = (b[k] - sigma) / A[k][k]
   - Untuk indeks ke-i, di mana : 0 <= i < n
     tampilkan akar x[i]

Algoritma Eliminasi Gauss-Pivoting

(asumsi : indeks array selalu dimulai dari 0)

1. Masukkan jml ordo matriks n dan inputkan masing-masing elemen augmented matriks A
2. Masukkan nilai epsilon (ep) untuk mentolerir nilai pivot

3. Lakukan pengecekan pivot mulai dari kolom ke-i, di mana : 0 <= i < n lakukan :
   - pivot = A[i][i]
   - besar = i;
   - jika |pivot| < ep, maka perlu dilakukan pertukaran baris sbb :
     - untuk baris ke-p, di mana : i+1 <= p < n lakukan
       jika |pivot| < |A[p][i]|, maka
       pivot = A[p][i]
       besar = p
     - tukar_baris ke- i dengan baris ke-besar

4. Lakukan pertukaran baris ke-i dengan baris ke-besar
   - Untuk kolom ke-j, di mana : 0 <= j <= n lakukan :
     - temp[j] = A[i][j]
     - A[i][j] = A[besar][j]
     - A[besar][j] = temp[j]

5. Untuk kolom ke-i, di mana : 0 <= i < n-1 lakukan OBE (operasi baris elementer) sbb :
   Untuk baris ke-j, di mana: i+1 <= j < n
   i. hitung nilai konstanta c : c = A[j][i] / A[i][i]
   ii. Untuk kolom ke-k, di mana : 0 <= k < n+1 (termasuk kolom yg berisi vektor b) hitunglah :
   temp = c * A[i][k]
   A[j][k] = A[j][k] - temp

6. Isikan elemen matriks pada kolom ke-n (kolom terakhir dari augmented matriks) ke dalam matriks vektor b sbb :
   Untuk baris ke-i : 0 <= i < n
   b[i] = A[i][n]

7. Hitung akar x dengan cara melakukan substitusi mundur sbb :
   - x[n-1] = b[n-1]/A[n-1][n-1]
   - Untuk baris ke-k, di mana : n-2 >= k >= 0
     - sigma = 0
     - untuk kolom ke-j, di mana : k+1 <= j < n , hitunglah
       sigma = sigma + A[k][j] * x[j]
     - x[k] = (b[k] - sigma) / A[k][k]
   - Untuk indeks ke-i, di mana : 0 <= i < n
     tampilkan akar x[i]

<u>*Catatan:*</u>

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

Metode ini merupakan pengembangan metode eliminasi Gauss, hanya saja augmented matrik, pada sebelah kiri diubah menjadi matrik diagonal sebagai berikut:

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

Penyelesaian dari persamaan linier simultan diatas adalah nilai $d_1,d_2,d_3,\dots,d_n$ dan atau:

$$x_1 = d_1, x_2 = d_2, x_3 = d_3, \dots, x_n = d_n$$

Teknik yang digunakan dalam metode eliminasi Gauss-Jordan ini sama seperti metode eliminasi Gauss yaitu menggunakan OBE (Operasi Baris Elementer). Hanya perhitungan penyelesaian secara langsung diperoleh dari nilai pada kolom terakhir dari setiap baris.

Algoritma Metode Eliminasi Gauss-Jordan adalah sebagai berikut:

(asumsi : indeks array selalu dimulai dari 0)

1. Masukkan jml ordo matriks n dan inputkan masing-masing elemen augmented matriks A

2. Masukkan nilai epsilon (ep) untuk mentolerir nilai pivot

3. Lakukan pengecekan pivot mulai dari kolom ke-i, di mana : 0 <= i < n lakukan :
   a. pivot = A[i][i]
   b. besar = i;
   c. jika |pivot| < ep, maka perlu dilakukan pertukaran baris sbb :
      - untuk baris ke-p, di mana : i+1 <= p < n lakukan
        jika |pivot| < |A[p][i]|, maka
        pivot = A[p][i]
        besar = p
      - tukar_baris ke- i dengan baris ke-besar

4. Lakukan pertukaran baris ke-i dengan baris ke-besar
   - Untuk kolom ke-j, di mana : 0 <= j <= n lakukan :
     - temp[j] = A[i][j]
     - A[i][j] = A[besar][j]
     - A[besar][j] = temp[j]

5. Untuk kolom ke-i, di mana : 0 <= i < n-1 lakukan OBE (operasi baris elementer) pada baris-baris di BAWAH diagonal sbb :
   Untuk baris ke-j, di mana: i+1 <= j < n
   i. hitung nilai konstanta c : c = A[j][i] / A[i][i]
   ii. Untuk kolom ke-k, di mana : 0 <= k < n+1 (termasuk kolom yg berisi vektor b) hitunglah :
   temp = c * A[i][k]
   A[j][k] = A[j][k] - temp

6. Untuk baris ke-j, di mana : 0 <= j < n lakukan operasi untuk menjadikan semua elemen pada diagonal bernilai 1 sbb :
   pivot = A[j][j]
   Untuk kolom ke-k, di mana : 0 <= k < n hitunglah :
   A[j][k] = A[j][k] / pivot

7. Untuk kolom ke-i, di mana : 0 <= i < n lakukan OBE (operasi baris elementer) pada baris-baris di ATAS diagonal sbb :
   Untuk baris ke-j, di mana: i-1 >= j >= 0
   i. hitung nilai konstanta c : c = A[j][i]
   ii. Untuk kolom ke-k, di mana : 0 <= k <= n+1 (termasuk kolom yg berisi vektor b) hitunglah :
   temp = c * A[i][k]
   A[j][k] = A[j][k] - temp

8. Hitung akar x dengan cara melakukan substitusi mundur sbb :

- Untuk indeks ke-i, di mana : 0 <= i < n
  tampilkan akar x[i] = A[i][n]

(1) Masukkan matrik A, dan vektor B beserta ukurannya n

(2) Buat augmented matrik [A|B] namakan dengan A

(3) Untuk baris ke i dimana i=1 s/d n

(a) Perhatikan apakah nilai $a_{i,i}$ sama dengan nol :

Bila ya :

pertukarkan baris ke i dan baris ke i+k≤n, dimana $a_{i+k,i}$ tidak sama dengan nol, bila tidak ada berarti perhitungan tidak bisa dilanjutkan dan proses dihentikan dengan tanpa penyelesaian.

Bila tidak : lanjutkan

(b) Jadikan nilai diagonalnya menjadi satu, dengan cara untuk setiap kolom k dimana k=1 s/d n+1, hitung $a_{i,k} = \dfrac{a_{i,k}}{a_{i,i}}$

(1) Untuk baris ke j, dimana j = i+1 s/d n

Lakukan operasi baris elementer: untuk kolom k dimana k=1 s/d n

Hitung $c = a_{j,i}$

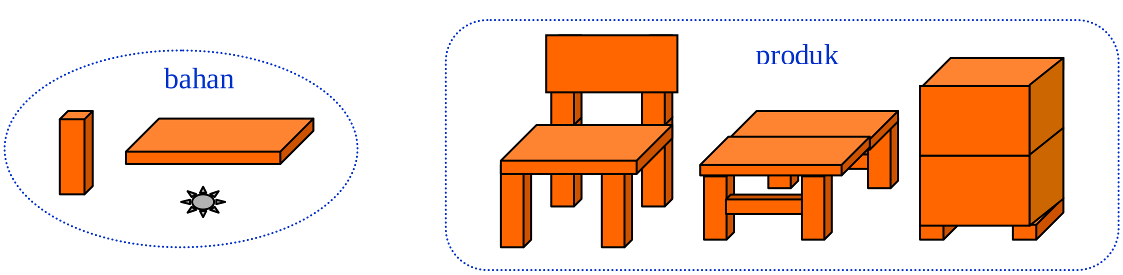
Hitung $a_{j,k} = a_{j,k} - c.a_{i,k}$

(2) Penyelesaian, untuk i = n s/d 1 (bergerak dari baris ke n sampai baris pertama)

$x_i = a_{i,n+1}$

C. TUGAS PENDAHULUAN

Sebuah industri membuat tiga macam produk yaitu kursi, meja dan lemari. Produk produk tersebut membutuhkan tiga jenis bahan yaitu kayu papan, kayu ring dan paku penguat. Perhatikan contoh produknya sebagai berikut :



Spesifikasi produk:

- 1 kursi membutuhkan 2 kayu papan, 6 ring dan 10 paku.
- 1 meja membutuhkan 2 kayu papan, 6 ring dan 12 paku
- 1 lemari membutuhkan 10 kayu papan, 10 ring dan 20 paku

Berapa jumlah meja, kursi dan lemari yang dapat dibuat bila tersedia 108 kayu papan, 204 kayu ring dan 376 paku ?

Gunakan metode gauss, gauss Jordan dan gauss seidel secara manual.

D. PERCOBAAN

1. Implementasikan algoritma dan flowchart yang sudah diberikan dan dikerjakan pada laporan pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan
2. Jalankan program, kemudian tampilkan, tuliskan augmented matrik dan hasil akhir penyelesaian persamaan linier simultan prosedur no 1.
3. Lakukan penukaran baris matrik persamaan linier simultan : baris II dengan baris III pada matrik awal yang diketahui. Jalankan program kemudian tampilkan, tuliskan augmented matrik dan hasil akhir penyelesaian persamaan linier simultan dari matrik yang telah ditukar barisnya.
4. Apa pengaruh dari penukaran baris pada matrik prosedur 4.

E. LAPORAN RESMI

FORM LAPORAN AKHIR
Nama dan NRP mahasiswa

Judul Percobaan : METODE ELIMINASI GAUSS Jordan

Algoritma :

Listing program yang sudah benar :

Hasil percobaan :

1. Augmented matrik asal :
2. Percobaan dilakukan dengan : MAX_ITER=___ dan e=____
3. Untuk nilai awal = (___,___,___)

n	x1(n)	x2(n)	x3(n)	e

Dilakukan minimal 4 kali dengan 4 nilai awal yang berbeda

4. Penyelesaian akhir persamaan linier simultan :
   - x1 = ….
   - x2 = ….
   - x3 = ….
5. Ulangi langkah 2 s/d 4 untuk matrik penukaran baris, kemudian lakukan untuk matrik penukaran kolom

Apa pengaruh dari pertukaran baris matrik persamaan linier simultan :

PRAKTIKUM 5

PERSAMAAN LINIER SIMULTAN 2

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan
- Metode Gauss Seidel
- Pivoting
- Contoh Kasus Persamaan Linier Simultan

B. DASAR TEORI

Metode interasi Gauss-Seidel adalah metode yang menggunakan proses iterasi hingga diperoleh nilai-nilai yang berubah. Bila diketahui persamaan linier simultan:

$$\begin{array}{ccccccccccccccc} a_{11} & x_1 & + & a_{12} & x_2 & + & a_{13} & x_3 & + & ... & + & a_{1n} & x_n & = & b_1 \\ a_{21} & x_1 & + & a_{22} & x_2 & + & a_{23} & x_3 & + & ... & + & a_{2n} & x_n & = & b_2 \\ a_{31} & x_1 & + & a_{32} & x_2 & + & a_{33} & x_3 & + & ... & + & a_{3n} & x_n & = & b_3 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ a_{n1} & x_1 & + & a_{n2} & x_2 & + & a_{n3} & x_3 & + & ... & + & a_{nn} & x_n & = & b_n \end{array}$$

Berikan nilai awal dari setiap x_i (i=1 s/d n) kemudian persamaan linier simultan diatas dituliskan menjadi:

$$x_1 = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2 - a_{13}x_3 - .... - a_{1n}x_n\right)$$

$$x_2 = \frac{1}{a_{2^2}}\left(b_2 - a_{21}x_1 - a_{23}x_3 - .... - a_{2n}x_n\right)$$

....

$$x_n = \frac{1}{a_{nn}}\left(b_n - a_{n1}x_1 - a_{n2}x_2 - .... - a_{nn-1}x_{n-1}\right)$$

Dengan menghitung nilai-nilai x_i (i=1 s/d n) menggunakan 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 persamaan linier simultan tersebut. Atau dengan kata lain proses iterasi dihentikan bila selisih nilai x_i(i=1 s/d n) dengan nilai x_i pada iterasi sebelumnya kurang dari nilai tolerasi error yang ditentukan.

<u>Catatan:</u>

Hati-hati dalam menyusun sistem persamaan linier ketika menggunakan metode iterasi Gauss-Seidel ini. Perhatikan setiap koefisien dari masing-masing x_i pada semua persamaan di diagonal utama (a_{ii}). Letakkan nilai-nilai terbesar dari koefisien untuk setiap x_i pada diagonal utama. Masalah ini adalah '**masalah pivoting**' yang harus benar-benar diperhatikan, karena penyusun yang salah akan menyebabkan iterasi menjadi divergen dan tidak diperoleh hasil yang benar.

Algoritma Metode Iterasi Gauss-Seidel adalah sebagai berikut:

Algoritma Eliminasi Gauss-Jordan
(asumsi : indeks array selalu dimulai dari 0)

1. Masukkan jml ordo matriks n dan inputkan masing-masing elemen augmented matriks A

2. Masukkan nilai epsilon (ep) untuk mentolerir nilai pivot

3. Lakukan pengecekan pivot mulai dari kolom ke-i, di mana : 0 <= i < n lakukan :
   a. pivot = A[i][i]
   b. besar = i;
   c. jika |pivot| < ep, maka perlu dilakukan pertukaran baris sbb :
      - untuk baris ke-p, di mana : i+1 <= p < n lakukan
        jika |pivot| < |A[p][i]|, maka
          pivot = A[p][i]
          besar = p
      - tukar_baris ke- i dengan baris ke-besar

4. Lakukan pertukaran baris ke-i dengan baris ke-besar
   - Untuk kolom ke-j, di mana : 0 <= j <= n lakukan :
     - temp[j] = A[i][j]
     - A[i][j] = A[besar][j]
     - A[besar][j] = temp[j]

5. Untuk kolom ke-i, di mana : 0 <= i < n-1 lakukan OBE (operasi baris elementer) pada baris-baris di BAWAH diagonal sbb :
   Untuk baris ke-j, di mana: i+1 <= j < n
     i. hitung nilai konstanta c : c = A[j][i] / A[i][i]

ii. Untuk kolom ke-k, di mana : 0 <= k < n+1 (termasuk kolom yg berisi vektor b) hitunglah :

temp = c * A[i][k]

A[j][k] = A[j][k] - temp

6. Untuk baris ke-j, di mana : 0 <= j < n lakukan operasi untuk menjadikan semua elemen pada diagonal bernilai 1 sbb :

pivot = A[j][j]

Untuk kolom ke-k, di mana : 0 <= k < n hitunglah :

A[j][k] = A[j][k] / pivot

7. Untuk kolom ke-i, di mana : 0 <= i < n lakukan OBE (operasi baris elementer) pada baris-baris di ATAS diagonal sbb :

Untuk baris ke-j, di mana: i-1 >= j >= 0

i. hitung nilai konstanta c : c = A[j][i]

ii. Untuk kolom ke-k, di mana : 0 <= k <= n+1 (termasuk kolom yg berisi vektor b) hitunglah :

temp = c * A[i][k]

A[j][k] = A[j][k] - temp

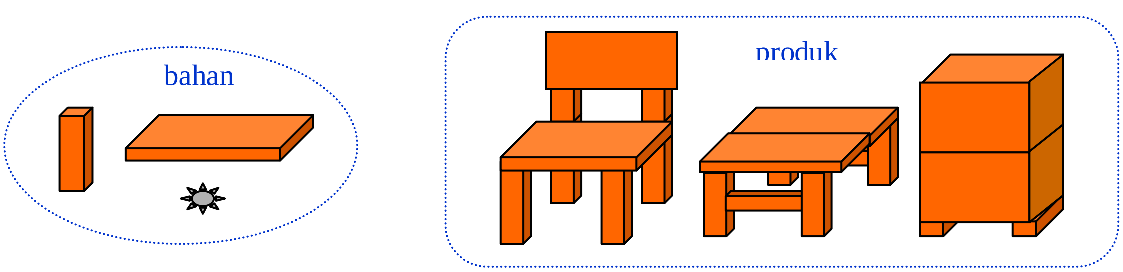
8. Hitung akar x dengan cara melakukan substitusi mundur sbb :

- Untuk indeks ke-i, di mana : 0 <= i < n

tampilkan akar x[i] = A[i][n]

C. TUGAS PENDAHULUAN

Sebuah industri membuat tiga macam produk yaitu kursi, meja dan lemari. Produk produk tersebut membutuhkan tiga jenis bahan yaitu kayu papan, kayu ring dan paku penguat. Perhatikan contoh produknya sebagai berikut :



Spesifikasi produk:

- 1 kursi membutuhkan 2 kayu papan, 6 ring dan 10 paku.
- 1 meja membutuhkan 2 kayu papan, 6 ring dan 12 paku

- 1 lemari membutuhkan 10 kayu papan, 10 ring dan 20 paku

Berapa jumlah meja, kursi dan lemari yang dapat dibuat bila tersedia 108 kayu papan, 204 kayu ring dan 376 paku ?

Gunakan metode gauss, gauss Jordan dan gauss seidel secara manual.

D. PERCOBAAN

1. Implementasikan algoritma dan flowchart yang sudah diberikan dan dikerjakan pada laporan pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan
2. Jalankan program, kemudian tampilkan, tuliskan augmented matrik dan hasil akhir penyelesaian persamaan linier simultan prosedur no 1.
3. Lakukan penukaran baris matrik persamaan linier simultan : baris II dengan baris III pada matrik awal yang diketahui. Jalankan program kemudian tampilkan, tuliskan augmented matrik dan hasil akhir penyelesaian persamaan linier simultan dari matrik yang telah ditukar barisnya.
4. Apa pengaruh dari penukaran baris pada matrik prosedur 4.

E. LAPORAN RESMI

FORM LAPORAN AKHIR
Nama dan NRP mahasiswa

Judul Percobaan : METODE ELIMINASI GAUSS SEIDEL

Algor

Listing program yang sudah benar :

Hasil percobaan :

5. Augmented matrik asal :
6. Percobaan dilakukan dengan : MAX_ITER=___ dan e=____
7. Untuk nilai awal = (___,___,___)

n	x1(n)	x2(n)	x3(n)	e

Dilakukan minimal 4 kali dengan 4 nilai awal yang berbeda

8. Penyelesaian akhir persamaan linier simultan :
   - x1 = ….
   - x2 = ….
   - x3 = ….
5. Ulangi langkah 2 s/d 4 untuk matrik penukaran baris, kemudian lakukan untuk matrik penukaran kolom

Apa pengaruh dari Pertukaran baris tersebut :

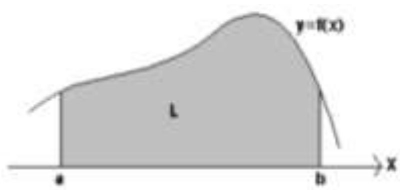
PRAKTIKUM 6
INTEGRASI

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan
- Metode Integrasi Reimann
- Metode Trapezoida
- Metode Simpson

B. DASAR TEORI

Perhitungan integral adalah perhitungan dasar yang digunakan dalam kalkulus untuk banyak keperluan. Integral ini secara definitive digunakan untuk menghitung luasdaerah yang dibatasi oleh fungsi y = f(x) dan sumbu x. Perhatikan gambar berikut:



Gambar 6.1 Luas daerah kurva

Luas daerah yang diarsir L dapat dihitung dengan :

$$L = \int_a^b f(x)dx$$

Pada beberapa permasalahan perhitungan integral ini, dapat dihitung secara manualdengan mudah, tetapi pada banyak permasalahan integral sulit sekali dihitung bahkan dapat dikatakan tidak dapat dihitung secara manual.

Pada penerapannya, perhitungan integral ini digunakan untuk menghitung luas area pada peta, volume permukaan tanah, menghitung luas dan volume-volume benda putar dimana fungsi f(x) tidak ditulis, hanya digunakan gambar untuk menyajikan nilai f(x). Sebagai contoh, diketahui foto daerah sebagai berikut:

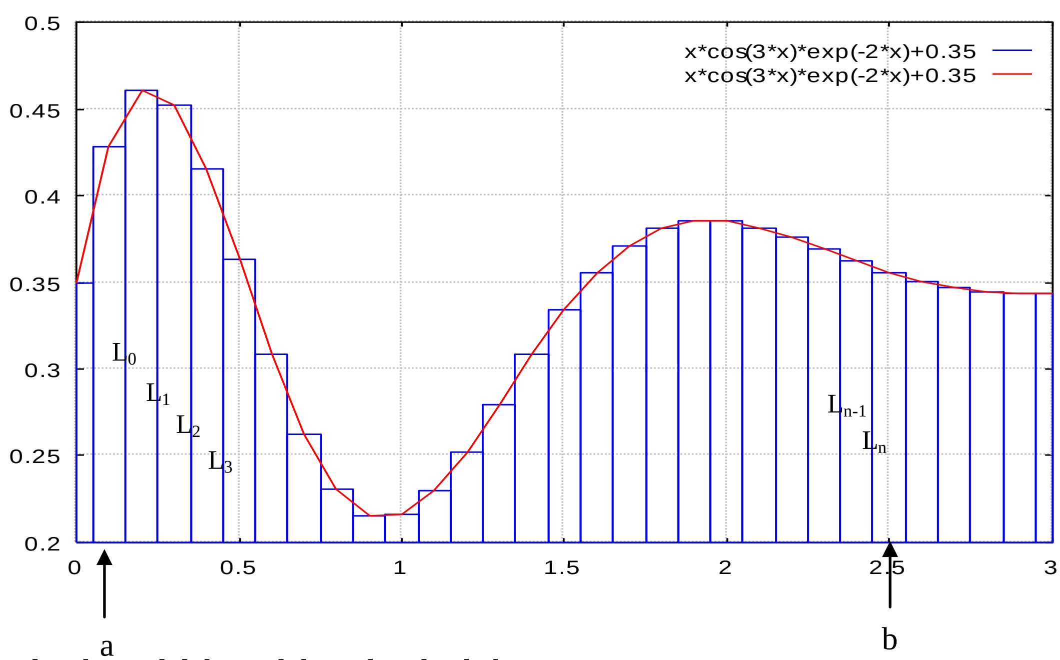
Gambar 6.2 Luas daerah

Untuk menghitung luas daerah yang diarsir L, perlu digunakan analisa numerik.Karena polanya disajikan dalam gambar dengan faktor skala tertentu.

Metode integral Reimann ini merupakan metode integral yang digunakan dalam kalkulus, dan didefinisikan dengan:

$$\int_a^b f(x)dx = \lim_{x \to 0} \sum_{i=0}^{n} f(x_i)$$

Pada metode ini, luasan yang dibatasi oleh y = f(x) dan sumbu x dibagi menjadi N bagian pada range x = $[a,b]$ yang akan dihitung.Kemudian dihitung tinggi dari setiap 3 tep ke-I yaitu $f(x_i)$.L_i adalah luas setiap persegi panjang dimana $L_i=f(x_i).\,\Delta x_i$



Luas keseluruhan adalah jumlah L_i dan dituliskan :

$$
\begin{aligned}
L &= L_0 + L_1 + L_2 + .. + L_n \\
&= f(x_0)\Delta x_0 + f(x_1)\Delta x_1 + f(x_2)\Delta x_2 + ... + f(x_n)\Delta x_3 \\
&= \sum_{i=0}^{n} f(x_i)\Delta x_i
\end{aligned}
$$

Bila diambil $\Delta x_0 = \Delta x_1 = \Delta x_2 = ... = \Delta x_n = L$ maka didapat metode integral reimam sebagai berikut :

$$
\int_a^b f(x)dx = h\sum_{i=0}^{n} f(x_i)
$$

Algoritma Metode Integral Reimann:

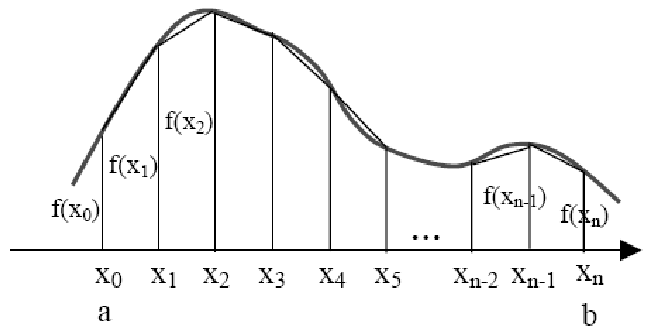
(1) Definisikan fungsi f(x)
(2) Tentukan batas bawah dan batas ata integrasi
(3) Tentukan jumlah pembagi area N
(4) Hitung h=(b-a)/N
(5) Hitung $L = h.\sum_{i=0}^{N} f(x_i)$

Tabel 6.1 perhitungan integral dengan metode Reimann

	A	B	C	D	E	F	G	H
1	i	x	f(x)	Eksak				
2	0	0	1	-0.4		a	0	
3	1	0.1	1.003158	-0.36192		b	1	
4	2	0.2	1.005716	-0.32742		h	0.1	
5	3	0.3	1.007758	-0.29618				
6	4	0.4	1.009359	-0.26789				
7	5	0.5	1.010585	-0.24228				
8	6	0.6	1.011493	-0.21909				
9	7	0.7	1.012133	-0.1981				
10	8	0.8	1.012546	-0.1791				
11	9	0.9	1.012771	-0.16191				
12	10	1	1.012839	-0.14635				
13		sigma f(x)	11.09836	-2.80024	error			
14		h*sigma f(x)	1.109836	-0.28002	1.38986			
15								

Metode Trapezoida

Pada metode integral Reimann setiap daerah bagian dinyatakan sebagai empat persegi panjang dengan tinggi $f(x_i)$ dan lebar Δx_i. Pada metode trapezoida ini setiap bagian dinyatakan sebagai trapezium seperti pada gambar berikut:



Luas trapesium ke-i (L_i) adalah:

$$L_i = \frac{1}{2}\left(f(x_i) + f(x_{i+1})\right).\Delta x_i \text{ atau } L_i = \frac{1}{2}\left(f_i + f_{i+1}\right).\Delta x_i$$

Dan luas keseluruhan dihitung dengan menjumlahkan luas seluruh trapesium:

$$L = \sum_{i=0}^{n-1} L_i$$

Sehingga diperoleh:

$$L_i = \sum_{i=0}^{n-1} \frac{1}{2} h\left(f_i + f_{i+1}\right) = \frac{h}{2}\left(f_0 + 2f_1 + 2f_2 + ... + f_n\right)$$

Contoh excel Trapezoida f(x)=e^{-x}sin2x+1

Algoritma

(1) Definisikan fungsi f(x)
(2) Tentukan nilai batas bawah *a*, batas atas *b*, dan jumlah pembagi N
(3) Hitung $h = \frac{(b-a)}{N}$
(4) Untuk i=0 sampai dengan i=N hitung:
(5) Hitung $x = a + i \times h$
(6) Hitung *f(x)*
(7) Jika I terletak antara 0 dan N maka $2 \times f(x)$
(8) Jika i=0 atau i=N maka tidak dikalikan 2.
(9) Hitung $L = \frac{h}{2}\left(f_0 + 2\sum_{i=1}^{n-1} f_i + f_n\right)$
(10) Hitung nilai integral secara eksak / kalkulus
(11) Hitung nilai error = $|L - eksak|$
(12) Tampilkan nilai i, x, dan f(x) dalam bentuk tabel, tampilkan nilai L, eksak, dan error.

Tabel 6. 2 Metode Trapezoida

	A	B	C	D	E	F	G	H	I	J
1	i	x	f(x)	Eksak						
2	0	0	1	-0.4		a	0		n=	10
3	1	0.1	1.003158	-0.361915122		b	1			
4	2	0.2	1.005716	-0.327420478		h	0.1			
5	3	0.3	1.007758	-0.296181072						
6	4	0.4	1.009359	-0.267892835						
7	5	0.5	1.010585	-0.242279795						
8	6	0.6	1.011493	-0.219091515						
9	7	0.7	1.012133	-0.198100756						
10	8	0.8	1.012546	-0.179101351						
11	9	0.9	1.012771	-0.161906274						
12	10	1	1.012839	-0.146345882						
13		sigma f(x)	9.085519	-2.253889197	error					
14		integral =	1.009194	-0.252706214	1.261900079					
15										

Metode Simpson 1/3

Metode integrasi Simpson merupakan pengembangan metode integrasi trapezoida, hanya saja daerah pembaginya bukan berupa trapesium tetapi berupa dua buah trapesium dengan menggunakan pembobot berat di titik tengahnya seperti telihat pada gambar berikut ini. Atau dengan kata lain metode ini adalah metode rata-rata dengan pembobot kuadrat.

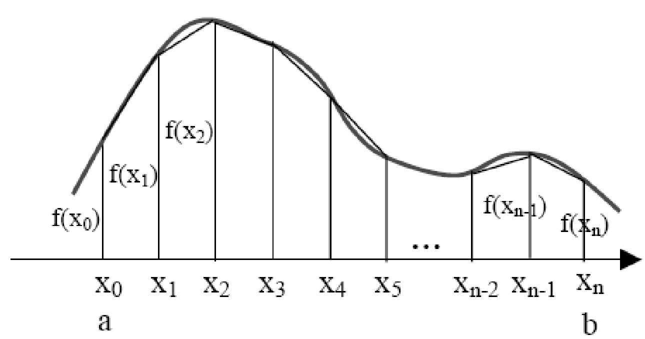
Bila menggunakan trapesium luas bangun di atas adalah:

$$L_i = \frac{h}{2}\left(f_{i-1} + f_i\right) + \frac{h}{2}\left(f_i + f_{i+1}\right) = \frac{h}{2}\left(f_{i-1} + 2f_i + f_{i+1}\right)$$

Pemakaian aturan simpson dimana bobot *fi* sebagai titik tengah dikalikan dengan 2 untuk menghitung luas bangun diatas dapat dituliskan dengan:

$$L_i = \frac{h}{3}\left(f_{i-1} + 2f_i\right) + \frac{h}{3}\left(2f_i + f_{i+1}\right) = \frac{h}{3}\left(f_{i-1} + 4f_i + f_{i+1}\right)$$

Perhatikan gambar berikut:



Dengan menggunakan aturan Simpson ini, luas dari daerah yang dibatas fungsi $y = f(x)$ dan sumbu *x* dapat dihitung sebagai berikut:

$$L_i = \frac{h}{3}\left(f_0 + 2f_1\right) + \frac{h}{3}\left(2f_1 + f_2\right) + \frac{h}{3}\left(f_2 + 2f_3\right) + \ldots + \frac{h}{3}\left(f_{n-2} + 2f_{n-1}\right) + \frac{h}{3}\left(2f_{n-1} + 2f_n\right)$$

Dapat dtuliskan dengan:

$$L = \frac{h}{3}\left(f_0 + 4\sum_{i=ganjil} f_i + 2\sum_{i=genap} f_i + f_n \right)$$

Contoh excel Simpson 1/3 $f(x)=e^{-x}\sin 2x+1$

Algoritma

(1) Definisikan fungsi f(x)

(2) Tentukan nilai batas bawah *a*, batas atas *b*, dan jumlah pembagi N

(3) Hitung $h = \frac{(b-a)}{N}$

(4) Untuk i=0 sampai dengan i=N hitung:

(5) Hitung $x = a + i \times h$

(6) Hitung *f(x)*

(7) Jika I terletak antara 0 dan N maka:

a. Jika i adalah genap maka $2\times f(x)$, jika tidak $4\times f(x)$

(8) Jika i=0 atau i=N maka *f*(*x*) tetap.

(9) Hitung $L = \frac{h}{3}\left(f_0 + 4\sum_{i=ganjil} f_i + 2\sum_{i=genap} f_i + f_n \right)$

(10) Hitung nilai integral secara eksak / kalkulus

(11) Hitung nilai error = $|L - eksak|$

Tampilkan nilai i, x, dan f(x) dalam bentuk tabel, tampilkan nilai L, eksak, dan error.

Tabel 6. 3 Metode Simpson

	A	B	C	D	E	F	G	H	I	J	K
1	i	x	f(x)	Eksak	ganjil	genap					
2	0	0	1	-0.4		1	a	0		n=	10
3	1	0.1	1.003158472	-0.361915122	1.003158472		b	1			
4	2	0.2	1.005715773	-0.327420478		1.005715773	h	0.1			
5	3	0.3	1.007757688	-0.296181072	1.007757688						
6	4	0.4	1.009359129	-0.267892835		1.009359129					
7	5	0.5	1.01058542	-0.242279795	1.01058542						
8	6	0.6	1.011493444	-0.219091515		1.011493444					
9	7	0.7	1.012132661	-0.198100756	1.012132661						
10	8	0.8	1.012546001	-0.179101351		1.012546001					
11	9	0.9	1.012770662	-0.161906274	1.012770662						
12	10	1	1.012838807	-0.146345882		1.012838807					
13		sigma f(x)	9.085519249	-2.253889197	5.046404902	4.039114347					
14		4*sigma f(x)			20.18561961	8.078228693					
15											
16		integral =	1.009222904	error							
17				0.755568785							

C. TUGAS PENDAHULUAN

Tuliskan dasar-dasar komputasi dari metode Reimann, Trapezoida, dan Simpson untuk menyelesaikan integrasi numerik, sebagai beriku : dasar teori, algoritma dan flowchart

D. PERCOBAAN

1. Didefinisikan suatu fungsi yang akan dicari nilai integralnya :

   $f(x) = \int_0^3 (3x^3 + 5x^2 - 10x^2 + 25)dx$

2. Implementasikan algoritma yang sudah diberikan dan dikerjakan pada laporan pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan
3. Jalankan program, dengan memasukkan berbagai macam nilai jumlah pembagi area (=Σbilah,=N), dan tuliskan semua hasil yang telah dicoba (ambil N=10, 20, 50, 100, 500 dan 1000)
4. Hitung pula nilai error dari selisih luasan eksak dan luasan dengan semua metode diatas
5. Apa pengaruh besal kecilnya nilai N terhadap error yang dihasilkan
6. Bandingkan ketiga metode tersebut.

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas , tambahkan dalam laporan resmi flow chart

FORM LAPORAN AKHIR
Nama dan NRP mahasiswa

Judul Percobaan :
Algoritma :

Listing program yang sudah benar :

Hasil percobaan :

9. Range batas bawah dan batas atas = [___ , ____]
10. Jumlah pembagi area N (=Σbilah) =______
11. Nilai L luasan dengan Metode Simpson = _____
12. Nilai L luasan eksak (kalkulus) = ____
13. Nilai e error = ______

No 1 s/d 5 diulangi untuk N=10, 20, 50, 100, 500 dan 1000

PRAKTIKUM 7

DIFFERENSIASI NUMERIK

A. TUJUAN PEMBELAJARAN

Memahami dan mampu menerapkan
- Selisih Maju
- Selisih Tengah
- Selisih Tengahan

B. DASAR TEORI

Persoalan menghitung turunan fungsi cukup banyak muncul dalam bidang rekayasa. Misalnya dalam bidang pengolahan citra (*image processing*), turunan fungsi diterapkan untuk mendeteksi sisi (*edge*) obyek pada suatu citra (lihat bagian terakhir bab ini). Sementara dalam perhitungan numerik sendiri, turunan fungsi dalam orde yang lebih tinggi, f', f'', f''', ..., kadang-kadang diperlukan. Misalnya untuk menghitung batas-batas galat interpolasi polinom dengan rumus

$$E(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\ldots(x-x_n)f^{(n+1)}(x)}{(n+1)!} \quad , x_0 \le \xi \le x_n$$

atau untuk menghitung galat integrasi numerik dengan aturan trapesium :

$$E(x) = \frac{-1}{12}(b\text{-}a)\, h^2 f''(t) \, , \qquad a \le t \le b$$

Bila persamaan fungsi $f(x)$ diberikan secara eksplisit, maka kita dapat menentukan fungsi turunannya, $f'(x)$, $f''(x)$, ..., $f\ (n+1)\ (x)$, lalu menggunakannya untuk menghitung nilai turunan fungsi di $x = t$.

Seringkali fungsi $f(x)$ tidak diketahui secara eksplisit, tetapi kita hanya memiliki beberapa titik data saja. Pada kasus seperti ini kita tidak dapat menemukan nilai turunan fungsi secara analitik. Sebaliknya, pada kasus lain, meskipun *f*(*x*) diketahui secara eksplisit tetapi bentuknya rumit sehingga menentukan fungsi turunannya merupakan pekerjaan yang tidak mangkus dan tidak praktis, misalnya pada fungsi-fungsi berikut ini:

(i) $f(x) = \dfrac{\sqrt{\cos(2x^2) + x\tan(3x)}}{\sin(x) + e^x - 2x/\cos(x)}$,

(ii) $f(x) = x\, e^{(2x+2)}\, ln\,(4x^2)$,

(iii) dan sebagainya.

Untuk kedua kasus terakhir, perhitungan nilai turunan dapat dikerjakan secara numerik (*numerical differentiation* atau *numerical derivative*). Nilai turunan yang diperoleh merupakan nilai hampiran. Sebagaimana halnya pada integrasi numerik, perhitungan turunan numerik juga menggunakan nilai-nilai diskrit. Karena itu, fungsi dalam bentuk tabel merupakan bentuk alami untuk perhitungan turunan.

PERSOALAN TURUNAN NUMERIK

Persoalan turunan numerik ialah menentukan hampiran nilai turunan fungsi *f* yang diberikan dalam bentuk tabel. Meskipun metode numerik untuk menghitung turunan fungsi tersedia, tetapi perhitungan turunan sedapat mungkin dihindari. Alasannya, nilai turunan numerik umumnya kurang teliti dibandingkan dengan nilai fungsinya. Dalam kenyataannya, turunan adalah limit dari hasil bagi selisih: yaitu pengurangan dua buah nilai yang besar (*f*(*x*+*h*) - *f*(*x*)) dan membaginya dengan bilangan yang kecil (*h*). Pembagian ini dapat menghasilkan turunan dengan galat yang besar. Lagi pula, jika fungsi *f* dihampiri oleh polinom interpolasi *p*, selisih nilai fungsi mungkin kecil tetapi turunannya boleh jadi sangat berbeda dengan nilai turunan sejatinya. Hal ini masuk akal sebab turunan numerik bersifat "halus", dan ini berlawanan dengan integrasi numerik, yang tidak banyak dipengaruhi oleh ketidaktelitian nilai fungsi, karena integrasi pada dasarnya adalah proses penghalusan [KRE88].

Metode selisih maju

Untuk menghitung differensial dapat digunakan metode selisih maju yaitu :

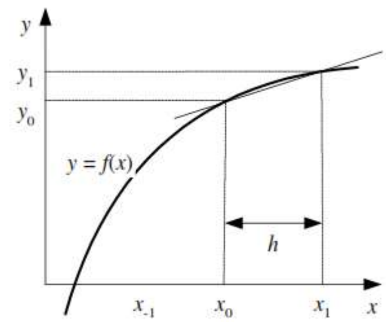
$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Untuk menghitung differensial ini maka diambil h yang kecil, karena error dari metode ini :

$$E(f) = -\frac{1}{2} h f^{11}(x)$$

Gambar Metode Selisih maju

$$f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} = \frac{f_1 - f_0}{h}$$



Algoritma Selisih Maju adalah sebagai berikut:

(1) Definisikan fungsi f(x) yang akan dicari nilai turunannya

(2) Definisikan fungsi turunan f'eksak(x) sebenarnya

(3) Masukkan nilai pendekatan awal : batas bawah a, batas atas b, dan nilai step h

(4) Untuk x=a sampai dengan b hitung :

$$f'(x) = \frac{f(x+h)-f(x)}{h}$$

(5) Tampilkan nilai x, f(x), f'(x) dan f'eksak(x)

Metode selisih tengahan merupakan metode pengambilan perubahan dari dua titik sekitar dari titik yang diukur.Perhatikan selisih maju pada titik x-h adalah :

$$f_1^1(x-h) = \frac{f(x)-f(x-h)}{h}$$

Dan selisih maju pada titik x adalah :

$$f_2^1(x) = \frac{f(x+h)-f(x)}{h}$$

Metode selisih tengahan merupakan rata-rata dari dua selisih maju :

$$f^{1}(x) = \frac{f_1^{1}(x) + f_2^{1(x)}}{2}$$

Atau dituliskan :

$$f^{1}(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Kesalahan pada metode ini adalah :

$$E(f) = -\frac{h^2}{6} f^{111}(\eta)$$

Tabel 7.1 metode maju

x	f(x)	ft_n	f eksak	error
0	1	1.797634443	2	
0.1	1.179763444	1.390653283	1.593838379	0.203185096
0.2	1.318828773	0.994686598	1.18937315	0.194686552
0.3	1.418297432	0.625607354	0.804549891	0.178942537
0.4	1.480858168	0.295197837	0.453174779	0.157976942
0.5	1.510377952	0.011359441	0.145041877	0.133682435
0.6	1.511513896	-0.221540421	-0.113781592	0.10775883
0.7	1.489359854	-0.402224819	-0.320553483	0.081671336
0.8	1.449137372	-0.532004717	-0.475377754	0.056626963
0.9	1.3959369	-0.614250707	-0.580683857	0.03356685
1	1.334511829		-0.640695561	0.640695561

Column1	Column2
a	0
b	1
h	0.1

Selisih Tengahan

Metode selisih tengahan merupakan metode pengambilan perubahan dari dua titik sekitar dari titik yang diukur.Perhatikan selisih maju pada titik x-h adalah:

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} = \frac{f_1 - f_{-1}}{2h}$$

Metode selisih tengahan merupakan rata-rata dari dua selisih maju :

$$f^{1}(x) = \frac{f_1^{1}(x) + f_2^{1(x)}}{2}$$

Atau dituliskan:

$$f^{1}(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Error pada metode ini adalah:

$$E(f) = -\frac{h^2}{6} f^{111}(\eta)$$

Algoritma Selisih Tengah adalah sebagai berikut

(1) Definisikan fungsi f(x) yang akan dicari nilai turunannya

(2) Definisikan fungsi turunan f'eksak(x) sebenarnya

(3) Masukkan nilai pendekatan awal : batas bawah a, batas atas b, dan nilai step h

(4) Untuk x=a sampai dengan b hitung :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

(4) Tampilkan nilai x, f(x), f'(x) dan f'eksak(x)

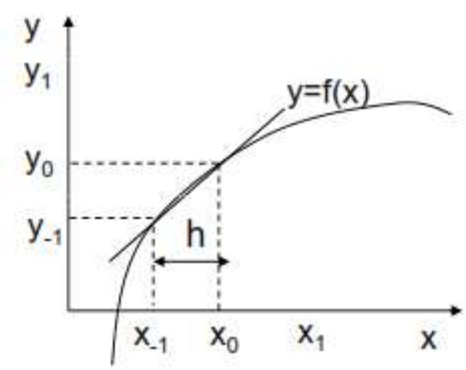
Tabel 7. 2 Metode Tengahan

x	f(x)	f_n	f_eksak	error
0	1	1.996635055	2	0.003364945
0.1	1.179763444	1.594143863	1.593838379	0.000305484
0.2	1.318828773	1.192669941	1.18937315	0.003296791
0.3	1.418297432	0.810146976	0.804549891	0.005597085
0.4	1.480858168	0.460402595	0.453174779	0.007227816
0.5	1.510377952	0.153278639	0.145041877	0.008236762
0.6	1.511513896	-0.10509049	-0.113781592	0.008691101
0.7	1.489359854	-0.31188262	-0.320553483	0.008670863
0.8	1.449137372	-0.467114768	-0.475377754	0.008262986
0.9	1.3959369	-0.573127712	-0.580683857	0.007556145

a	0
b	1
h	0.1

Selisih Mundur

Metode selisih mundur merupakan metode pengambilan titik kebalikan dari metode maju. Perhatikan selisih mundur pada titik x-h adalah :



Metode selisih tengahan merupakan rata-rata dari dua selisih maju :

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

atau

$$f'(x) \approx \frac{f_0 - f_{-1}}{h}$$

Error pada metode ini adalah:

$$E(f) = -\frac{1}{2} h f''(x)$$

Algoritma Selisih Mundur adalah sebagai berikut:

1. Definisikan fungsi f(x) yang akan dicari nilai turunannya.
2. Definisikan fungsi turunan f'eksak(x) sebenarnya.
3. Masukkan nilai pendekatan awal : batas bawah a, batas atas b, dan nilai step h .
4. Untuk x=a sampai dengan b hitung :

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

5. Tampilkan nilai x, f(x), f'(x) dan f'eksak(x)

Tabel 7.3 Mundur

x	f(x)	f_n	f_eksak	error
0	1	2.195635667	2	0.195635667
0.1	1.179763444	1.797634443	1.593838379	0.617870999
0.2	1.318828773	1.390653283	1.18937315	0.071824511
0.3	1.418297432	0.994686598	0.804549891	0.423610834
0.4	1.480858168	0.625607354	0.453174779	0.855250814
0.5	1.510377952	0.295197837	0.145041877	1.215180115
0.6	1.511513896	0.011359441	-0.113781592	1.500154454
0.7	1.489359854	-0.221540421	-0.320553483	1.710900275
0.8	1.449137372	-0.402224819	-0.475377754	1.851362191
0.9	1.3959369	-0.532004717	-0.580683857	1.927941617
1	1.334511829	-0.614250707	-0.640695561	1.948762536

a	0
b	1
h	0.1

C. TUGAS PENDAHULUAN

Tuliskan dasar-dasar komputasi dari ketiga metode diatas untuk menyelesaikan differensiasi numerik, sebagai berikut :

1.Judul : METODE

2. Dasar teori dari metode
3. Algoritma dan Flowchart

D. PERCOBAAN

1. Didefinisikan suatu fungsi yang akan dicari nilai differensialnya :

   $f(x)=e^{-x}sin(2x)+1$

2. Implementasikan algoritma yang sudah diberikan dan dikerjakan pada laporan pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan
3. Jalankan program, dengan memasukkan berbagai macam nilai h dan tulislah semua hasil yang telah dicoba (h=0.1|0.01|0.001|0.0001)
4. Hitung pula nilai error dari selisih nilai fungsi turunan eksak dan nilai fungsi turunan selisih Mundur, diakhir iterasi dapatkan rata-rata errornya
5. Apa pengaruh besar kecilnya nilai h terhadap nilai rata-rata error no.4

E. LAPORAN RESMI

FORM LAPORAN AKHIR
Nama dan NRP mahasiswa

Judul Percobaan : METODE

Algoritma

Listing program yang sudah benar :

Hasil percobaan :

14. Range batas bawah dan batas atas = [____,_____]
15. Interval h =_______

(Dilakukan minimal 4 kali)

n	f(x)	f'(x)	f'eksak(x)	error

PRAKTIKUM 8

PENYELESAIAN KASUS

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan Review dari Pertemuan 1 sampai dengan 7

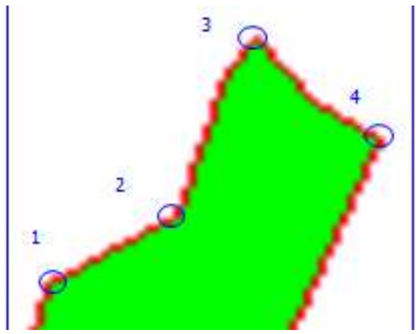
B. TUGAS PENDAHULUAN

Pelajari materi dari pratikum 1 samapai dengan 7

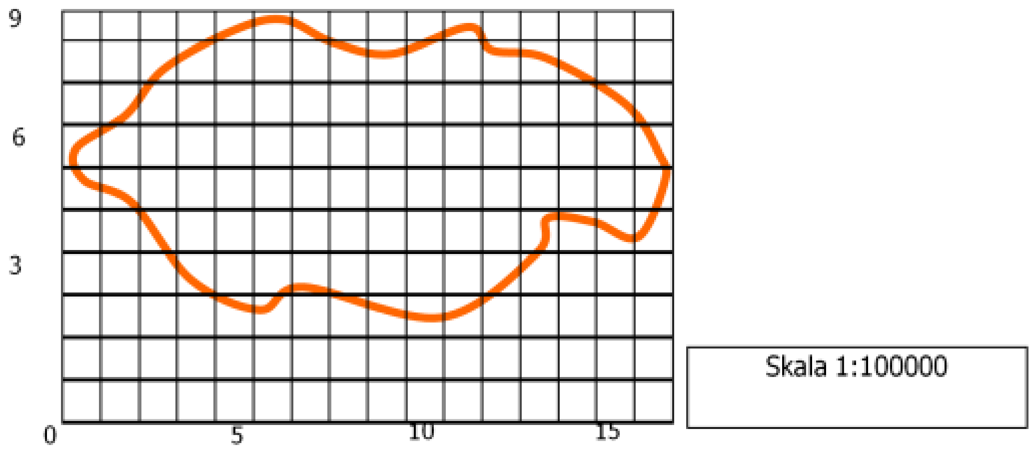
C. PERCOBAAN

1. Sebuah sinyal DTMF mempunyai persamaan : sin(x)+sin(2x). Tentukan nilai maksimal dari sinyal tersebut untuk batas 0 s/d 3, menggunakan metode Secant.
2. Tentukan titik potong kurva y = e-x dengan $y=x^2$ untuk batas [-1,1].

   Gunakan metode Secant dan Newton Raphson. Bandingkan jumlah iterasi dan kesalahannya.
3. Gunakan metode Newton Raphson, Regula Falsi dan Secant untuk menghitung akar dari 10. Bandingkan jumlah iterasi dan kesalahannya. Buat kesimpulan.
4. Perhatikan bahwa pada ke-4 titik tersebut dihubungkan dengan garis lurus, sehingga tampak kasar. Untuk menghaluskannya dilakukan pendekatan garis dengan kurva yang dibentuk dengan fungsi pendekatan polinomial. Dari fungsi polinomial yang dihasilkan kurva dapat digambarkan dengan lebih halus. Misalkan pada contoh diatas, 4 titik yang ditunjuk adalah (2,3), (7,6), (8,14) dan (12,10). 4 titik ini dapat didekati dengan fungsi polinom pangkat 3



5. Tentukan luas dari



n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
y(n)	0	1	2.5	4.5	6	7	6.5	6	6	6.5	6.5	6	5.5	3.5	3	3	0

D. LAPORAN RESMI

Kumpulkan hasil percobaan di atas , tambahkan dalam laporan resmi flow chart

PRAKTIKUM 9

PERASAMAAN DEFFRENSIAL BIASA 1

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan Ploting dari persamaan Differensial

B. DASAR TEORI

Persamaan diferensial dapat dibedakan menjadi dua macam tergantung pada jumlah variabel bebas. Apabila persamaan tersebut mengandung hanya satu variabel bebas, persamaan disebut dengan persamaan diferensial parsial. Derajat (order) dari persamaan ditentukan oleh derajat tertinggi dari turunannya.

Sebagai contoh persamaan diferensial biasa di bawah ini adalah berorder satu, karena turunan tertingginya adalah turunan pertama.

$$x\frac{dy}{dx} + y = 3$$

Sedang persamaan diferensial biasa berorder dua mengandung turunan kedua sebagai turunan tertingginya, seperti bentuk di bawah ini:

$$\frac{d^2y}{dx^2} + 3\frac{dx}{dy} + 2y = 0$$

Contoh persamaan diferensial parsial dengan variabel bebas *x* dan *t* adalah:

$$\frac{\partial y}{\partial t} = \frac{\partial^2 y}{\partial x^2}$$

Penyelesaian persamaan diferensial adalah suatu fungsi yang memenuhi persamaan diferensial dan juga memenuhi kondisi awal yang diberikan pada persamaan tersebut. Di dalam penyelesaian persamaan diferensial secara analitis, biasanya dicari penyelesaian umum yang mengandung konstanta sembarang dan kemudian mengevaluasi konstanta tersebut sedemikian sehingga hasilnya sesuai dengan kondisi awal. Metode penyelesaian persamaan diferensial secara analitis terbatas pada persamaan-persamaan dengan bentuk tertentu, dan biasanya hanya untuk menyelesaikan persamaan linier dengan koefisien konstan.

Misalkan suatu persamaan diferensial biasa berorder satu, sebagai berikut:

$$\frac{dy}{dx} = y \tag{9.1}$$

Penyelesaian dari persamaan tersebut adalah:

$$y = C\,e^{x} \tag{9.2}$$

yang memberikan banyak fungsi untuk berbagai nilai koefisien *C*. Gambar 8.1, menunjukkan beberapa kemungkinan dari penyelesaian persamaan (8.2), yang tergantung pada nilai *C*.

Untuk mendapatkan penyelesaian tunggal diperlukan informasi tambahan, misalnya nilai y (x) dan atau turunannya pada nilai x tertentu. Untuk persamaan order n biasanya diperlukan n kondisi untuk mendapatkan penyelesaian tunggal y (x). Apabila semua n kondisi diberikan pada nilai x yang sama (misalnya x_0), maka permasalahan disebut dengan problem nilai awal. Apabila dilibatkan lebih dari satu nilai x, permasalahan disebut dengan problem nilai batas. Misalnya persamaan (8.1), disertai kondisi awal yaitu $x = 0$, nilai $y = 1$ atau:

$$y(x = 0) = 1 \tag{9.3}$$

Substitusikan persamaan (9.3) ke dalam persamaan (9.2) memberikan:

$$1 = C\,e^{0}$$

atau

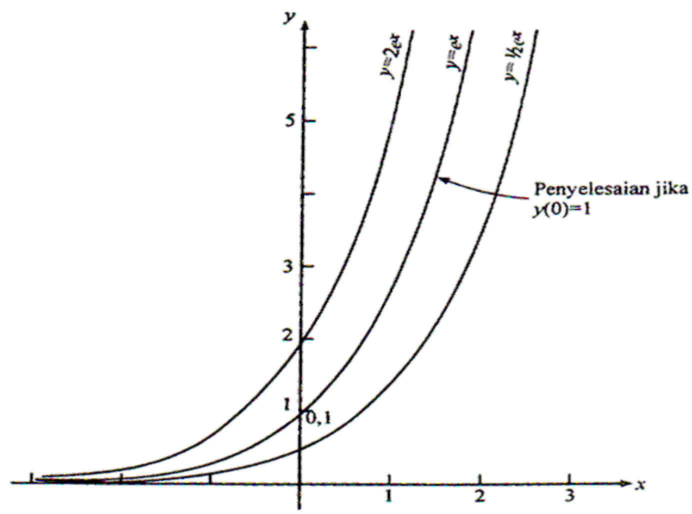
$$C = 1$$

Dengan demikian penyelesaian tunggal yang memenuhi persamaan:

$$\frac{dy}{dx} = y$$

$$y(x = 0) = 1$$

adalah:

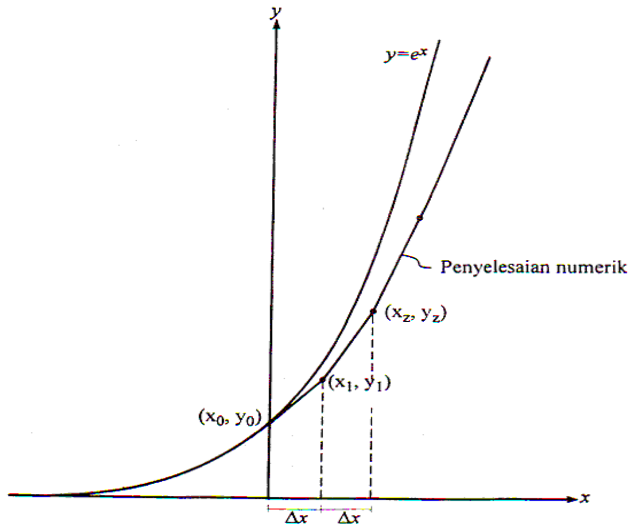


$y = e^x$

Gambar 9.1. Penyelesaian persamaan $\frac{dy}{dx} = y$

Metode penyelesaian numerik tidak ada batasan mengenai bentuk persamaan diferensial. Penyelesaian berupa tabel nilai-nilai numerik dari fungsi untuk berbagai variabel bebas. Penyelesaian suatu persamaan diferensial dilakukan pada titik-titik yang ditentukan secara berurutan. Untuk mendapatkan hasil yang lebih teliti maka jarak (interval) antara titik-titik yang berurutan tersebut dibuat semakin kecil.

Penyelesaian persamaan (9.1) dan persamaan (9.3) adalah mencari nilai y sebagai fungsi dari x. Persamaan diferensial memberikan kemiringan kurve pada setiap titik sebagai fungsi x dan y. Hitungan dimulai dari nilai awal yang diketahui, misalnya di titik (x_0, y_0). Kemudian dihitung kemiringan kurve (garis singgung) di titik tersebut. Berdasar nilai y_0 di titik x_0 dan kemiringan fungsi di titik-titik tersebut dapat dihitung nilai y_1 di titik x_1 yang berjarak Δx dari x_0. Selanjutnya titik (x_1, y_1) yang telah diperoleh tersebut digunakan untuk menghitung nilai y_2 di titik x_2 yang berjarak Δx dari x_1. Prosedur hitungan tersebut diulangi lagi untuk mendapatkan nilai y selanjutnya, seperti pada Gambar 9.2.



Gambar 9.2. Penyelesaian numerik persamaan diferensial

Dalam Matlab, diferensial untuk fungsi polinom adalah relatif mudah. Misalnya $f(x) = x^5 + 2x^4 + 5x^2 + 7x + 3$ maka ambilah koefisien koefisiennya.

Contoh:

```
>> g=[1 2 5 7 3]
g =
1 2 5 7 3
>> h=polyder(g)
h =
4 6 10 7
```

Bentuk-bentuk deferensial lain juga bisa diperoleh apalag jika menggunakan *symbolyc math toolbox*. Tapi tidak setiap matlab dilengkapi dengan toolbox ini. Namun itu tidak masalah, kita akan coba membuat sendiri penyelesaiannya dengan memanfaatkan deret Taylor.

- **Diferensial Numerik**

```
function q=diffgen(func,n,x,h);
if ((n=1)|(n==2)|(n==3)|(n==4))
c=zeros(4,7);
c(1,:)=[ 0 1 -8 0 8 -1 0];
c(2,:)=[0 -1 16 -30 16 -1 0];
```

```
c(3,:)=[1.5 -12 19.5 0 -19.5 12 -1.5];
c(4,:)=[-2 24 -78 112 -78 24 -2];
y=feval(func,x+ [-3:3]*h);
q=c(n,:)*y' ; q = q/(12*h^n);
else
disp('n harus 1, 2, 3 atau 4 ');break
end
```

Penggunaan fungsi diatas:

Jika kita mempunya y = cos(x) dan kita akan menghitung turunan kedua dengan x = 1.2 dengan h atau ketelitian 0.01 maka dituliskan:

```
>> hasil=diffgen('cos',2,1.2,.01)
hasil =
-0.3624
```

Jika kita ingin menghitung sebuah diferensial disuatu titik maka kita harus mendefinisikan fungsinya terlebih dahulu.

C. TUGAS PENDAHULUAN

Pelajari tentang persamaan diffrensial biasa

D. PERCOBAAN

Lakukan percobaan diatas dengan mathlab

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas

PRAKTIKUM 10

PERASAMAAN DEFFRENSIAL BIASA 2

A. TUJUAN PEMBELAJARAN

Memahami dan mampu menerapkan
- Metode Euler
- Metode Taylor

Ploting

B. DASAR TEORI

Bentuk persamaan differensial berikut:

$y' = f(x, y)$ dan y adalah fungsi dalam x.

Dengan menggunakan pendekatan nilai awal (x_0,y_0) maka nilai-nilai y berikutnya daat diperoleh dengan:

$$y_{n+1} = y_n + h.f(x_n, y_n)$$

Dengan demikian dapat dinyatakan bahwa:

(1) Masukan yang diperlukan dalam menyelesaikan persamaan differensial dengan menggunakan metode Euler adalah :
- Nilai pendekatan awal (x_0,y_0)
- Jumlah iterasi N
- Step h

(2) Keluaran yang dihasilkan oleh metode ini adalah nilai-nilai y pada setiap x yang bertambah dengan step h.

(3) Bila x dan y ditabelkan akan dapat dibentuk grafik dari hasil penyelesaian persamaan differensial tersebut.

Algoritma dari Metode Euler:

(1) Definisikan model dari persamaan differensial dalam f(x,y)

(2) Masukkan nilai pendekatan awal x(0) dan y(0)

(3) Masukkan nilai maksimum iterasi N dan nilai step h.

(4) Untuk n=0 sampai dengan N

- x(n)=x0+h*n
- y(n+1)=y(n)+h*f(x(n),y(n))

(5) Tampilkan nilai x(n) dan y(n) dalam table untuk n=0 s/d N

Metode Taylor adalah suatu metode pendekatan yang menggunakan deret Taylor sebagai bentuk perbaikan nilai untuk nilai fungsi secara keseluruhan pada penyelesaian persamaan differensial. Dengan memberikan nilai pendekatan awal (x_0,y_0), penyelesaian dapat diperoleh dengan:

$$y(x) = y_0 + (x - x_0)y'(x_0) + \frac{(x - x_0)^2}{2!} y''(x_0) + \ldots + \frac{(x - x_0)^k}{k!} y^{(k)}(x_0)$$

dimana :

$$y' = f(x, y)$$
$$y'' = f'(x, y) = f_x + f_y y'$$
$$y^{(3)} = f''(x, y) = f'_x + f''_y y'$$
....
$$y^{(n)} = f^{(n-1)}(x, y) = f_x^{(n-2)} + f_y^{(n-2)} y'$$

Algoritma dari Metode Taylor:

(1) Definisikan model dari persamaan differensial dalam f(x,y)

(2) Definisikan model turunan ke n dari f(x,y)

(3) Masukkan nilai pendekatan awal x(0) dan y(0)

(4) Masukkan nilai maksimum iterasi N dan nilai step h.

(5) Masukkan jumlah suku maksimum K dari deret Taylor yang digunakan

(6) Untuk n=1 sampai dengan N

- x(n)=x0+h*n

- $y(n) = y0 + \sum_{k=1}^{K} \frac{(x(n)-x0)^k}{k!} f^{(k)}(x0, y0)$

(7) Tampilkan nilai x(n) dan y(n) dalam table untuk n=0 s/d N

C. TUGAS PENDAHULUAN

Tuliskan dasar-dasar komputasi dari metode Euler dan taylor untuk menyelesaikan persamaan differensil sebagai berikut:

1. Judul :
2. Dasar teori dari metode
3. Algoritma dan flowchart

D. PERCOBAAN

1. Didefinisikan persoalan dari persamaan differensial dengan fungsi sebagai berikut:

   $$\frac{dy}{dx} + 2y = 1$$

2. Tuliskan listing program yang sudah dibuat pada tugas pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan.
3. Jalankan program dengan memasukkan berbagai macam nilai awal dan tuliskan semua hasil yang telah dicoba
4. Bandingkan hasilnya dengan menggunakan penyelesaian analitik $y = \frac{1}{2}\left(1 - e^{-2x}\right)$
5. Seleseikan dengan kedua metode.
6. Apa pengaruh nilai awal terhadap penyelesaian persamaan differensial.

E. LAPORAN RESMI

FORM LAPORAN AKHIR

Nama dan NRP mahasiswa

Judul percobaan : METODE

Algoritma

Listing program yang sudah benar

Hasil percobaan :

Percobaan dilakukan dengan h = ___ dan N = _____

Untuk nilai awal = (____ , _____)

n	x(n)	y(n)	e(n)

(Dilakukan minimal 4 kali)

Error diukur dengan menbandingkan hasil komputasi dengan hasil perhitungan analitik. Rata-rata selisih Error adalah ______________

Pengaruh nilai awal pada penyelesaian persamaan differensial adalah

PRAKTIKUM 11

PERASAMAAN DEFFRENSIAL BIASA 3

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan Metode Runge Kutta

B. DASAR TEORI

Metode Runge Kutta 2

Metode Runge-Kutta membuat step yang lebih kecil dari perubahan nilai dengan membagi nilai perubahan tiap step menjadi sejumlah bagian yang ditentukan, bentuk paling sederhana dari metode Runge Kutta ini adalah membagi bagian perubahan menjadi dua bagian sehingga :

$$dy = \frac{h.f_1 + h.f_2}{2}$$

dimana f1 dan f2 adalah nilai fungsi step yang diambil dari bentuk fungsi persamaan differensial pada step tengahan.

Sehingga diperoleh formulasi dari Metode Runge-Kutta 2 sebagai berikut:

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

dimana: $k_1 = h.f(x_n, y_n)$

$$k_2 = h.f(x_n + h, y_n + k_1)$$

Algoritma Metode Runge Kutta 2:

(1) Definisikan model dari persamaan differensial dalam f(x,y)

(2) Masukkan nilai pendekatan awal x(0) dan y(0)

(3) Masukkan nilai maksimum iterasi N dan nilai step h.

(4) Untuk n=1 sampai dengan N

- x(n)=x0+h*n
- $k1 = h.f(x(n), y(n))$
- $k2 = h.f(x(n) + h, y(n) + k1)$
- $y(n+1) = y(n) + \frac{1}{2}(k1 + k2)$

(5) Tampilkan nilai x(n) dan y(n) dalam table untuk n=0 s/d N

Metode Runge Kutta 4

Bila pada metode Runge-Kutta 4, nilai koefisien perbaikannya adalah 4 buah, maka pada metode ini menggunakan 4 nilai koefisien perbaikan yaitu k_1, k_2, k_3, k_4 yang diberikan sebagai berikut:

$$y_{n+1} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

dimana : $k_1 = h.f(x_n, y_n)$

$$k_2 = h.f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = h.f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = h.f\left(x_n + h, y_n + k_3\right)$$

Algoritma dari Metode Runge Kutta 4:

(1) Definisikan model dari persamaan differensial dalam f(x,y)

(2) Masukkan nilai pendekatan awal x(0) dan y(0)

(3) Masukkan nilai maksimum iterasi N dan nilai step h.

(4) Untuk n=1 sampai dengan N

- x(n)=x0+h*n

- $k1 = h.f(x(n), y(n))$
- $k2 = h.f(x(n) + h/2, y(n) + k1/2)$
- $k3 = h.f(x(n) + h/2, y(n) + k2/2)$
- $k4 = h.f(x(n) + h, y(n) + k3)$
- $y(n+1) = y(n) + \frac{1}{6}(k1 + 2.k2 + 2.k3 + k4)$

(6) Tampilkan nilai x(n) dan y(n) dalam table untuk n=0 s/d N

C. Tugas Pendahuluan

Tuliskan dasar-dasar komputasi dari metode Runge Kutta 2 untuk menyelesaikan persamaan differensil sebagai berikut:

(1) Judul : METODE RUNGE KUTTA 4

(2) Dasar teori dari metode Euler

(3) Algoritma dan flowchart

D. Percobaan

1. Didefinisikan persoalan dari persamaan differensial dengan fungsi sebagai berikut:
   a. $\frac{dy}{dx} + 2y = 1$
2. Tuliskan listing program yang sudah dibuat pada tugas pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan.
3. Jalankan program dengan memasukkan berbagai macam nilai awal dan tuliskan semua hasil yang telah dicoba
4. Jalankan program dengan memasukkan berbagai nilai K, dan tuliskan semua hasil yang telah dicoba
5. Bandingkan hasilnya dengan menggunakan penyelesaian analitik $y = \frac{1}{2}\left(1 - e^{-2x}\right)$
6. Apa pengaruh nilai awal terhadap penyelesaian persamaan differensial.

PRAKTIKUM 12

PERSAMAAN DIFFERENSIAL TINGKAT TINGGI

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan Metode Runge Kutta

B. DASAR TEORI

Metode Runge Kutta 2

Perhatikan persamaan differensial tingkat 2 berikut:

$$\frac{d^2y}{dx^2} = F\left(x, y, \frac{dy}{dx}\right)$$

Ubah variabel: *y=y* dan *z=y'* sehingga diperoleh 2 persamaan differensial tingkat satu berikut:

$$y' = z$$
$$z' = F(x, y, z)$$

ini berarti diperoleh 2 fungsi masing-masing:

$$f(x, y, z) = z$$
$$g(x, y, z) = F(x, y, z)$$

Dengan menggunakan metode Runge-Kutta 2 diperoleh:

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

$$z_{n+1} = z_n + \frac{1}{2}(l_1 + l_2)$$

dimana:

$$k_1 = h.f(x, y, z)$$

$$l_1 = h.g(x, y, z)$$

$$k_2 = h.f(x + h, y + k_1, z + l_1)$$

$$l_2 = h.g(x + h, y + k_1, z + l_1)$$

Algoritma Metode Runge Kutta 2:

(1) Definisikan model dari persamaan differensial dalam f(x,y,z)

(2) Masukkan nilai pendekatan awal x(0), y(0) dan z(0) dimana z(0)=y'(0)

(3) Masukkan nilai maksimum iterasi N dan nilai step h.

(4) Untuk n=0 sampai dengan N

- x(n)=x0+h*n
- $k1 = h.f\left(x(n), y(n), z(n)\right)$
- $l1 = h.g\left(x(n), y(n), z(n)\right)$
- $k2 = h.f\left(x(n) + h, y(n) + k1, z(n) + l1\right)$
- $l2 = h.g\left(x(n) + h, y(n) + k1, z(n) + l1\right)$

(5) Tampilkan nilai x(n), y(n) dan z(n) dalam table untuk n=0 s/d N

Metode Runge Kutta 4

Perhatikan persamaan differensial tingkat 2 berikut:

$$\frac{d^2y}{dx^2} = F\left(x, y, \frac{dy}{dx}\right)$$

Ubah variabel: *y=y* dan *z=y'* sehingga diperoleh 2 persamaan differensial tingkat satu berikut:

$$y' = z$$
$$z' = F(x, y, z)$$

ini berarti diperoleh 2 fungsi masing-masing:

$$f(x, y, z) = z$$
$$g(x, y, z) = F(x, y, z)$$

Dengan menggunakan metode Runge-Kutta 2 diperoleh:

$$y_{n+1} = y_n + \frac{1}{2}\left(k_1 + k_2\right)$$

$$z_{n+1} = z_n + \frac{1}{2}\left(l_1 + l_2\right)$$

dimana: $k_1 = h.f(x, y, z)$

$$l_1 = h.g(x, y, z)$$

$$k_2 = h.f(x + h, y + \frac{k_1}{2}, z + \frac{l_1}{2})$$

$$l_2 = h.g(x+h, y+k_1, z+l_1)$$

$$k_3 = h.f(x+h, y+\frac{k_2}{2}, z+\frac{l_2}{2})$$

$$l_3 = h.g(x+h, y+\frac{k_2}{2}, z+\frac{l_2}{2})$$

$$k_4 = h.f(x+h, y+k_3, z+l_3)$$

$$k_4 = h.g(x+h, y+k_3, z+l_3)$$

Algoritma Metode Runge Kutta 4:

(1) Definisikan model dari persamaan differensial dalam f(x,y,z)

(2) Masukkan nilai pendekatan awal x(0), y(0) dan z(0) dimana z(0)=y'(0)

(3) Masukkan nilai maksimum iterasi N dan nilai step h.

(4) Untuk n=0 sampai dengan N

- x(n)=x0+h*n
- $k1 = h.f\left(x(n), y(n), z(n)\right)$
- $l1 = h.g\left(x(n), y(n), z(n)\right)$
- $k2 = h.f\left(x(n)+h, y(n)+k1/2, z(n)+l1/2\right)$
- $l2 = h.g\left(x(n)+h, y(n)+k1/2, z(n)+l1/2\right)$
- $k3 = h.f\left(x(n)+h, y(n)+k2/2, z(n)+l2/2\right)$
- $l3 = h.g\left(x(n)+h, y(n)+k2/2, z(n)+l2/2\right)$
- $k4 = h.f\left(x(n)+h, y(n)+k3, z(n)+l3\right)$
- $l4 = h.g\left(x(n)+h, y(n)+k3, z(n)+l3\right)$

(4) Tampilkan nilai x(n), y(n) dan z(n) dalam table untuk n=0 s/d N

C. Pendahuluan

Tuliskan dasar-dasar komputasi dari metode Runge Kutta 4 untuk menyelesaikan persamaan differensil sebagai berikut:

(1) Judul : METODE RUNGE KUTTA 4 UNTUK MENYELESAIAKAN PERSAMAAN DIFFERENSIAL TINGKAT TINGGI

(2) Dasar teori dari metode Runge Kutta 4

(3) Algoritma dan flowchart

D. Percobaan

1. Didefinisikan persoalan dari persamaan differensial dengan fungsi sebagai berikut:

   $$\frac{d^2y}{dx^2} + y = 1$$

2. Tuliskan listing program yang sudah dibuat pada tugas pendahuluan, lalu isi lembaran laporan akhir seperti form laporan akhir yang ditentukan.
3. Jalankan program dengan memasukkan berbagai macam nilai awal dan tuliskan semua hasil yang telah dicoba
4. Bandingkan hasilnya dengan menggunakan penyelesaian analitik $y = 1 - \sin(x)$
5. Apa pengaruh nilai awal terhadap penyelesaian persamaan differensial.

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas , tambahkan dalam laporan resmi flow chart

PRAKTIKUM 13

INTERPOLASI_1

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan
- Linier
- Lagrange

Contoh Kasus Interpolasi

B. DASAR TEORI

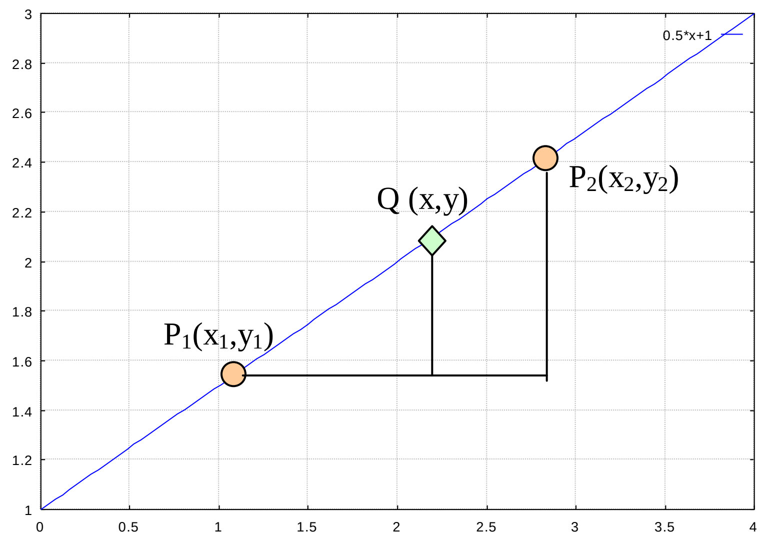
Mempelajari berbagai metode Interpolasi yang ada untuk menentukan titik-titik antara dari n buah titik dengan menggunakan suatu fungsi pendekatan tertentu. Metode Interpolasi yang dipelajari :

1. Interpolasi Linier
2. Interpolasi Kuadratik
3. Interpolasi Polinomial
4. Interpolasi Lagrange

Dasar Teori :

Interpolasi linier

Menentukan titik-titik antara dari 2 buah titik dengan menggunakan garis lurus.



Gambar 22.1. Kurva untuk interpolasi linier

Persamaan garis lurus yang melalui 2 titik $P_1(x_1,y_1)$ dan $P_2(x_2,y_2)$ dapat dituliskan dengan:

$$\frac{y-y_1}{y_2-y_1}=\frac{x-x_1}{x_2-x_1}$$

Sehingga diperoleh persamaan dari interpolasi linier sebagai berikut:

$$y=\frac{y_2-y_1}{x_2-x_1}(x-x_1)+y_1$$

Algoritma Interpolasi Linier:

(1) Tentukan dua titik P1 dan P2 dengan koordinatnya masing-masing (x1,y1) dan (x2,y2)

(2) Tentukan nilai x dari titik yang akan dicari

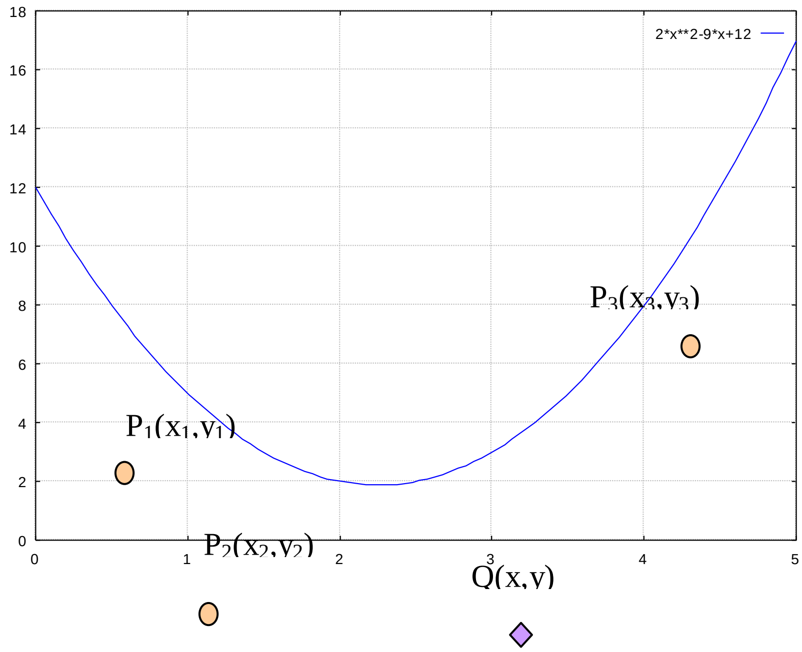
(3) Hitung nilai y dengan :

$$y=\frac{y_2-y_1}{x_2-x_1}(x-x_1)+y_1$$

(4) Tampilkan nilai titik yang baru Q(x,y)

Interpolasi Kuadratik

Interpolasi Kuadratik digunakan untuk mencari titik-titik antara dari 3 buah titik $P_1(x_1,y_1)$, $P_2(x_2,y_2)$ dan $P_3(x_3,y_3)$ dengan menggunakan pendekatan fungsi kuadrat.



Gambar 22.2. Kurva untuk interpolasi kuadratik

Untuk memperoleh titik Q(x,y) digunakan interpolasi kuadratik sebagai berikut:

$$y = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

Algoritma Interpolasi Kuadratik:

(1) Tentukan 3 titik input $P_1(x_1,y_1)$, $P_2(x_2,y_2)$ dan $P_3(x_3,y_3)$

(2) Tentukan nilai x dari titik yang akan dicari

(3) Hitung nilai y dari titik yang dicari menggunakan rumus dari interpolasi kuadratik:

$$y = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

(4) Tampilkan nilai x dan y

Interpolasi Polinomial

Interpolasi polynomial digunakan untuk mencari titik-titik antara dari n buah titik $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$, …, $P_N(x_N,y_N)$ dengan menggunakan pendekatan fungsi polynomial pangkat n-1:

$$y = a_0 + a_1 x + a_2 x^2 + ... + a_{n-1} x^{n-1}$$

Masukkan nilai dari setiap titik ke dalam persamaan polynomial di atas dan diperoleh persamaan simultan dengan n persamaan dan n variable bebas:

$$y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 + ... + a_{n-1} x_1^{n-1}$$

$$y_2 = a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 + ... + a_{n-1} x_2^{n-1}$$

$$y_3 = a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 + ... + a_{n-1} x_3^{n-1}$$

……………………………………………

$$y_n = a_0 + a_1 x_n + a_2 x_n^2 + a_3 x_n^3 + ... + a_{n-1} x_n^{n-1}$$

Penyelesaian persamaan simultan di atas adalah nilai-nilai a_0, a_1, a_2, a_3, …, a_n yang merupakan nilai-nilai koefisien dari fungsi pendekatan polynomial yang akan digunakan.

Dengan memasukkan nilai x dari titik yang dicari pada fungsi polinomialnya, akan diperoleh nilai y dari titik tersebut.

Algoritma Interpolasi Polynomial :

(1) Menentukan jumlah titik N yang diketahui.

(2) Memasukkan titik-titik yang diketahui $P_i = (x_i, y_i)$ untuk i=1,2,3,…,N

(3) Menyusun augmented matrik dari titik-titik yang diketahui sebagai berikut:

$$J = \left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & ... & x_1^{n-1} & y_1 \\ 1 & x_2 & x_2^2 & ... & x_2^{n-1} & y_2 \\ 1 & x_3 & x_3^2 & ... & x_3^{n-1} & y_3 \\ ... & ... & ... & ... & ... & ... \\ 1 & x_n & x_n^2 & ... & x_n^{n-1} & y_n \end{array}\right]$$

(4) Menyelesaikan persamaan simultan dengan augmented matrik di atas dengan menggunakan metode eliminasi gauss/Jordan.

(5) Menyusun koefisien fungsi polynomial berdasarkan penyelesaian persamaan simultan di atas.

$$a = \{a_i \mid a_i = J(i,n), 0 \le i \le n-1\}$$

(6) Memasukkan nilai x dari titik yang diketahui

(7) Menghitung nilai y dari fungsi polynomial yang dihasilkan

$$y = \sum_{i=0}^{N-1} a_i x^i$$

(8) Menampilkan titik (x,y)

Interpolasi Lagrange

Interpolasi polynomial digunakan untuk mencari titik-titik antara dari n buah titik $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$, …, $P_N(x_N,y_N)$ dengan menggunakan pendekatan fungsi polynomial yang disusun dalam kombinasi deret dan didefinisikan dengan:

$$y = \sum_{i=1}^{N} y_i \prod_{j \ne i} \frac{(x - x_j)}{(x_i - x_j)}$$

Algoritma Interpolasi Lagrange :

(1) Tentukan jumlah titik (N) yang diketahui

(2) Tentukan titik-titik $P_i(x_i,y_i)$ yang diketahui dengan i=1,2,3,…,N

(3) Tentukan x dari titik yang dicari

(4) Hitung nilai y dari titik yang dicari dengan formulasi interpolasi lagrange

$$y = \sum_{i=1}^{N} y_i \prod_{j \ne i} \frac{(x - x_j)}{(x_i - x_j)}$$

(5) Tampilkan nilai (x,y)

Diketahui 5 buah titik sebagai berikut:

n	x(n)	y(n)

1	1	4
2	4	2
3	6	3
4	7	5
5	10	8

Tentukan titik-titik pada x=2, 3, 5, 8, 9, 11 dan 12

Petunjuk:

(1) Tuliskan program dari flowchart yang sudah saudara buat pada tugas pendahuluan.

(2) Dengan menggunakan soal di atas, jalankan program dan masukkan nilai-nilai titik yang diketahui dan jumlah titiknya

(3) Masukkan nilai-nilai x dari titik-titik yang dicari

(4) Tampilkan hasil dari titik-titiknya.

(5) Simpan semua titik-titik baik yang diketahui maupun hasil perhitungan ke dalam file teks

(6) Dengan menggunakan gnuplot, tampilkan grafik dari file teks yang sudah dibuat.

E. LAPORAN RESMI

Kumpulkan hasil percobaan di atas , tambahkan dalam laporan resmi flow chart untuk menghitung nilai rata-rata dari n bilangan yang diinputkan, hitung jumlah totalnya, hitung maksimal dan minimal bilangan.

PRAKTIKUM 14

INTERPOLASI_1

Interpolasi Lagrange

Interpolasi polynomial digunakan untuk mencari titik-titik antara dari n buah titik $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$, …, $P_N(x_N,y_N)$ dengan menggunakan pendekatan fungsi polynomial yang disusun dalam kombinasi deret dan didefinisikan dengan:

$$y = \sum_{i=1}^{N} y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Algoritma Interpolasi Lagrange :

(6) Tentukan jumlah titik (N) yang diketahui

(7) Tentukan titik-titik $P_i(x_i,y_i)$ yang diketahui dengan i=1,2,3,…,N

(8) Tentukan x dari titik yang dicari

(9) Hitung nilai y dari titik yang dicari dengan formulasi interpolasi lagrange

$$y = \sum_{i=1}^{N} y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

(10) Tampilkan nilai (x,y)

Tugas Pendahuluan:

(1) Judul : Interpolasi Polinomial

(2) Dasar Teori

(3) Algoritma Dan Flowchart

Perhatikan soal berikut ini:

Diketahui 5 buah titik sebagai berikut:

n	x(n)	y(n)
1	1	4
2	4	2
3	6	3
4	7	5
5	10	8

Tentukan titik-titik pada x=2, 3, 5, 8, 9, 11 dan 12

Petunjuk:

(1) Tuliskan program dari flowchart yang sudah saudara buat pada tugas pendahuluan.

(2) Dengan menggunakan soal di atas, jalankan program dan masukkan nilai-nilai titik yang diketahui dan jumlah titiknya

(3) Masukkan nilai-nilai x dari titik-titik yang dicari

(4) Tampilkan hasil dari titik-titiknya.

(5) Simpan semua titik-titik baik yang diketahui maupun hasil perhitungan ke dalam file teks

(6) Dengan menggunakan gnuplot, tampilkan grafik dari file teks yang sudah dibuat.

Laporan Akhir

Judul : Interpolasi Lagrange

Algoritma dan Flowchart :

Listing Program:

Input :

n	x(n)	y(n)
1	1	4
2	4	2
3	6	3
4	7	5
5	10	8

Output :

x(n)	y(n)
1	4
2	
3	
4	2
5	
6	3

x(n)	y(n)
7	5
8	
9	
10	8
11	
12	

Hasil grafik:

PRAKTIKUM 15
REGRESI

A. TUJUAN PEMBELAJARAN

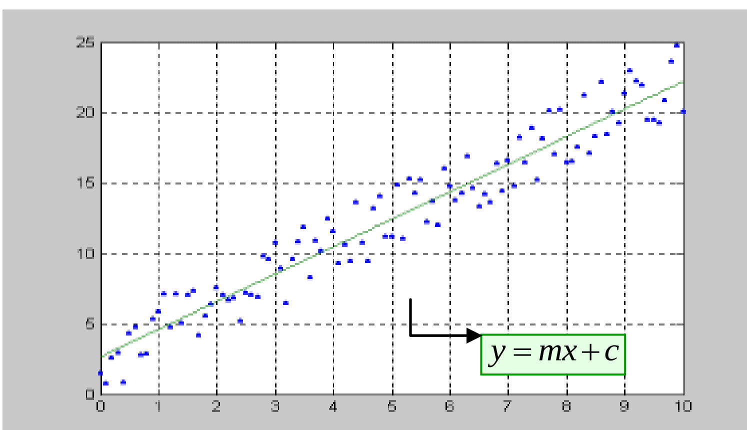
- Memahami dan mampu menerapkan
- Kurva fitting
- Linier
- Polinomial

Contoh Kasus Regresi

B. DASAR TEORI

Regresi Linier

Regresi linier digunakan menentukan fungsi linier (garis lurus) yang paling sesuai dengan kumpulan titik data (x_n,y_n) yang diketahui.



Gambar 23.1. Sebaran data dengan kurva linier

Dalam regresi linier ini yang dicari adalah nilai m dan c dari fungsi linier *y=mx+c*, dengan:

$$m = \frac{N\sum_{n=1}^{N} x_n y_n - \left(\sum_{n=1}^{N} x_n\right)\left(\sum_{n=1}^{N} y_n\right)}{N\sum_{n=1}^{N} x_n^2 - \left(\sum_{n=1}^{N} x_n\right)^2}$$

$$c = \frac{\sum_{n=1}^{N} y_n}{N} - m\frac{\sum_{n=1}^{N} x_n}{N} = \bar{y} - m\bar{x}$$

Algoritma regresi linier

(1) Tentukan N titik data yang diketahui dalam (x_i,y_i) untuk i=1,2,3,…,N

(2) Hitung nilai m dan c dengan menggunakan formulasi dari regresi linier di atas

(3) Tampilkan fungsi linier

(4) Hitung fungsi linier tersebut dalam range x dan step dx tertentu

(5) Tampilkan hasil tabel (x_n,y_n) dari hasil fungsi linier tersebut.

Regresi Eksponensial

Regresi eksponensial digunakan menentukan fungsi eksponensial yang paling sesuai dengan kumpulan titik data (x_n,y_n) yang diketahui. Regresi eksponensial ini merupakan pengembangan dari regresi linier dengan memanfaatkan fungsi logaritma.

Perhatikan :

$y = e^{-ax+b}$

dengan melogaritmakan persamaan di atas akan diperoleh:

$\ln y = \ln\left(e^{ax+b}\right)$

$\ln y = ax + b$

atau dapat dituliskan bahwa:

$z = ax + b$ dimana $z = \ln y$

Dengan demikian dapat digunakan regresi linier dalam menentukan fungsi eksponensial yang paling sesuai dengan data.

Algoritma regresi eksponensial

(1) Tentukan N titik data yang diketahui dalam (x_i,y_i) untuk i=1,2,3,…,N

(2) Ubah nilai y menjadi z dengan z = ln y

(3) Hitung nilai a dan b dengan menggunakan formulasi dari regresi linier di atas

(4) Tampilkan fungsi eksponensial $y = e^{-ax+b}$

(5) Hitung fungsi eksponensial tersebut dalam range x dan step dx tertentu

(6) Tampilkan hasil tabel (x_n,y_n) dari hasil fungsi eksponensial tersebut.

Regresi Polinomial

Regresi polinomial digunakan menentukan fungsi polynomial yang paling sesuai dengan kumpulan titik data (x_n,y_n) yang diketahui.

Fungsi pendekatan :

$$y = a_0 + a_1 x + a_1 x^2 + ... + a_n x^n$$

Regresi polinomial tingkat n dikembangkan dari model matrik normal sebagai berikut:

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^n & \sum_{i=1}^{n} x_i^{n+1} & \sum_{i=1}^{n} x_i^{n+2} & ... & \sum_{i=1}^{n} x_i^{2n} \\ \sum_{i=1}^{n} x_i^{n-1} & \sum_{i=1}^{n} x_i^{n} & \sum_{i=1}^{n} x_i^{n+1} & ... & \sum_{i=1}^{n} x_i^{2n-1} \\ \sum_{i=1}^{n} x_i^{n-2} & \sum_{i=1}^{n} x_i^{n-1} & \sum_{i=1}^{n} x_i^{n} & ... & \sum_{i=1}^{n} x_i^{2n-2} \\ ... & ... & ... & ... & ... \\ n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & ... & \sum_{i=1}^{n} x_i^n \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ ... \\ a_0 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i^n y_i \\ \sum_{i=1}^{n} x_i^{n-1} y_i \\ \sum_{i=1}^{n} x_i^{n-2} y_i \\ ... \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Hasil dari model matrik normal di atas adalah nilai-nilai a_0, a_1, a_2, ..., a_n.

Algoritma Regresi Polinomial

(1) Tentukan N titik data yang diketahui dalam (x_i,y_i) untuk i=1,2,3,…,N

(2) Hitung nilai-nilai yang berhubungan dengan jumlahan data untuk mengisi matrik normal

(3) Hitung nilai koefisien-koefisien a_0, a_1, a_2, ..., a_n dengan menggunakan eliminasi gauss/jordan

(4) Tampilkan fungsi polinomial $y = a_0 + a_1 x + a_1 x^2 + ... + a_n x^n$

(5) Hitung fungsi polinomial tersebut dalam range x dan step dx tertentu

(6) Tampilkan hasil tabel (x_n,y_n) dari hasil fungsi polinomial tersebut.

C. TUGAS PENDAHULUAN

(1) Judul: Regresi Polinomial

(2) Dasar Teori

(3) Algoritma
(4) Flowchart

1.1.1 Prosedur Percobaan

(1) Tuliskan program dari regresi polinomial sesuai dengan flowchart yang sudah dibuat pada tugas pendahuluan.
(2) Jalankan program dan isikan data-data sebagai berikut:

Jumlah produk	Keuntungan
5	10000
10	15000
15	18000
20	20000
25	25000
40	30000
45	40000
50	50000
55	70000
60	80000

(3) Tampilkan fungsi polinomial dari hasil regresi eksponensial.
(4) Tampilkan table dari fungsi hasil regresi polinomial pada x yang sama dengan data
(5) Tampilkan grafik fungsi polinomial yang dihasilkan.

Laporan Akhir

(1) Judul: Regresi Eksponensial
(2) Listing program
(3) Tuliskan tabel data di atas
(4) Tuliskan fungsi eksponensial hasil regresi eksponensial
(5) Gambarkan data dan garis hasil regresi
(6) Analisa

Jumlah produk	Keuntungan	Hasil Regresi	Error
5	10000		
10	15000		
15	18000		
20	20000		
25	25000		
40	30000		
45	40000		
50	50000		
55	70000		
60	80000		

(7) Hitung rata-rata error :

PRAKTIKUM 16

PENYELESEIAN STUDI KASUS

A. TUJUAN PEMBELAJARAN

- Memahami dan mampu menerapkan metode Numerik pada kasus Optimasi

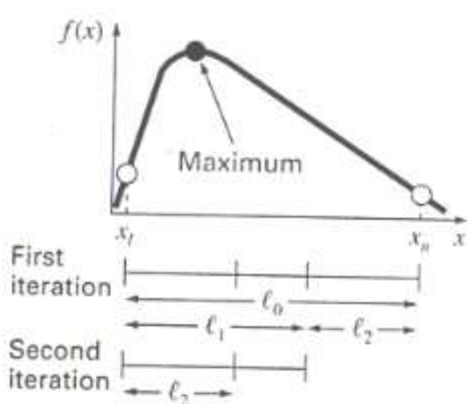
B. PERCOBAAN

Golden section merupakan salah satu cara atau metode optimasi numerik yang dapat diterapkan untuk fungsi yang bersifat unimodal . Kedua tipe optimasi, yaitu aksimasi dan minimasi dapat diselesaikan dengan cara ini. Golden-section (search) method merupakan metode optimasi satu variabel yang sederhana, dan mempunyai pendekatan yang mirip dengan metode bisection dalam penentuan akar persamaan tak linier.

Tinjaulah fungsi f(x) yang akan ditent

ukan maksimum-nya, pada rentang x = xl dan x = xu

(perhatikan gambar di bawah ini).



Mirip dengan bisection, ide dasar metode ini adalah memanf aatkan nilai yang lama sebagai nilai yang baru, secara iteratif. Sebagai akibatnya, rentang/ interval awal variabel yang dipilih semakin lama akan semakin menyempit, karena ada sebagian sub-interval variabel ya ng dieliminasi, hingga diperoleh tingkat konverg ensi yang diinginkan.

Gunakan metode *golden-section search* untuk menentukan maksimum dari fungsi:

$$f(x) = 2 \sin x - \frac{x^2}{10}$$

di dalam interval: $x_l = 0$ dan $x_u = 4$

C. LAPORAN RESMI

Kumpulkan hasil percobaan di atas , tambahkan dalam laporan resmi flow chart

DAFTAR PUSTAKA

1. Achmad Basuki, Nana Ramadijanti,”Pratikum Metode Numerik sebagai Algoritma komputasi Progra, Diploma IV”, modul ajar metode numerik,PENS ,2002.
2. Ardi Pujiyanta,”komputasi Numerik dengan Mathlab”,Graha ilmu,2007
3. http://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CCsQFjAB&url=http%3A%2F%2Felista.akprind.ac.id%2Fupload%2Ffiles%2F7778_Bab_8_1.doc&ei=fJuvUsCDLIWKrQfX2IDoCw&usg=AFQjCNFgT90FvKEJHBVl41Gql7c3LqQJ8Q&bvm=bv.57967247,d.bmk
4. http://diyarkholisoh.files.wordpress.com/2008/12/optimasi-numerik-doc-dy.pdf