



Solusi Numerik

Konstruksi Scribs & GUI Berbasis Matlab

Tidak semua permasalahan matematika dapat diselesaikan menggunakan metode analitik, namun butuh metode lain yang disebut metode numerik. Kemampuan metode numerik mendekati hasil eksak dilihat dari tingkat galat (error) yang diberikan. Permasalahannya adalah metode numerik membutuhkan bantuan software seperti Matlab dalam melakukan simulasi dan menemukan solusi yang tepat. Dalam prosesnya, kemampuan konstruksi scribs ataupun rancang bangun GUI sangat dibutuhkan karena melibatkan berbagai tingkat bahasa pemrograman baik struktur urut, percabangan, maupun perulangan. Buku ini hadir dengan memberikan warna baru dalam konstruksi scribs dan GUI berbasis Matlab, sehingga mempermudah dosen dan mahasiswa dalam menemukan solusi numerik dari persamaan dan sistem persamaan non linier yang diberikan. Selamat membaca!

www.buatbuku.com | [Penerbit Wade](#) | 0821-3954-7339 | redaksi@buatbuku.com



Solusi Numerik
Konstruksi Scribs & GUI Berbasis Matlab

Habib Ratu Perwira Negara, M.Pd. | Syaharuddin, M.Si.
Habibi Ratu Perwira Negara, M.Pd. | Kiki Riska Ayu Kurniawati, M.Pd.



Habib Ratu Perwira Negara, M.Pd.
Syaharuddin, M.Si.
Habibi Ratu Perwira Negara, M.Pd.
Kiki Riska Ayu Kurniawati, M.Pd.



Solusi Numerik

Konstruksi Scribs & GUI Berbasis Matlab

Solusi Numerik

Konstruksi Scribs & GUI Berbasis Matlab

Habib Ratu Perwira Negara, M.Pd

Syahrudin, M.Si

Habibi Ratu Perwira Negara, M.Pd

Kiki Riska Ayu Kurniawati, M.Pd

**Sanksi Pelanggaran Pasal 72
Undang-undang Nomor 19 Tahun 2002
Tentang Hak Cipta:**

1. Barangsiapa dengan sengaja dan tanpa hak mengumumkan atau memperbanyak ciptaan pencipta atau memberi izin untuk itu, dapat dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp. 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp. 5.000.000.000,00 (lima miliar rupiah).
2. Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta atau hak terkait, dapat dipidana dengan pidana paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).

Solusi Numerik

Konstruksi Scribs & GUI Berbasis Matlab

Solusi Numerik

Konstruksi Scribs & GUI Berbasis Matlab






© Habib Ratu Perwira Negara, M.Pd.
Syaharuddin, M.Si.
Habibi Ratu Perwira Negara, M.Pd.
Kiki Riska Ayu Kurniawati, M.Pd.

Editor : Vera Mandailina, M.Pd.
Malik Ibrahim, M.Pd.
Team WADE Publish
Layout : Team WADE Publish
Design Cover : Team WADE Publish

Diterbitkan oleh:



Jln. Pos Barat Km. 1 Melikan Ngimput Purwosari
Babadan Ponorogo Jawa Timur Indonesia 63491

 buatbuku.com
 redaksi@buatbuku.com
 0821-3954-7339
 Penerbit Wade
 [buatbuku](https://www.instagram.com/buatbuku)

Anggota IKAPI 182/JTI/2017

Cetakan Pertama, Desember 2018
ISBN: 978-623-7007-30-2

Hak Cipta dilindungi undang-undang.
Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa seizin tertulis dari Penerbit.

viii+88 hlm.; 15.5x23 cm

KATA PENGANTAR

Assalamu'alaikum Wr Wb

Alhamdulillah, segala puji hanya milik Allah SWT yang senantiasa memberikan kita nikmat-Nya yang tak terhingga, sehingga buku ajar “**Solusi Numerik: Konstruksi Scribs & GUI Berbasis Matlab**” ini bisa diselesaikan dengan baik. Harapan penulis, buku ajar ini bisa digunakan dengan baik dan bermanfaat bagi dosen dan mahasiswa baik saat perkuliahan maupun pengembangan kemampuan matematika komputasi kedepannya.

Buku ajar ini terbagi dalam tiga bab yakni:

1. **Konstruksi Scribs**. Mahasiswa dihadapkan pada layar kerja dan contoh penyelesaian konsep/persoalan matematika menggunakan bahasa dasar Matlab meliputi teknik penyelesaian persamaan non linier menggunakan metode *bisection*, *regula falsi*, *secant*, *Newton Raphson*, dan *Fixet Point*.
2. **GUI Matlab PNL & SPNL**. Mahasiswa diajarkan membuat sebuah *scribs* dari algoritma pemrograman untuk menyelesaikan sebuah konsep matematika menggunakan struktur urutan, percabangan, dan perulangan melalui penerapan perancangan GUI yang terdiri dari penyelesaian persamaan non linier dan sistem persamaan non linier menggunakan metode terbuka, tertutup, *Jacobian*, *Gauss Seidel*, dan *Newton Raphson*
3. **Rancang Bangun Aplikasi Project**. Mahasiswa diajarkan cara membuat tampilan desain *form* menggunakan atribut dan *tool* yang dimiliki oleh matlab. M-file yang sudah tersimpan akan dilanjutkan ke pembuatan *guide* agar penyelesaian solusi numerik lebih efisien, efektif dan menarik baik struktur urutan, perulangan, dan percabangan. Di samping itu, pada tahap ini mahasiswa diajarkan membuat aplikasi *project* yang merupakan aplikasi baru yang bisa *runtime* tanpa instalasi *set up* induk dari Matlab.



Mengingat begitu banyak wahana baru dalam isi buku ajar ini, tim penulis mengucapkan banyak terima kasih kepada Kemenristekdikti RI, LPPM STMIK Bumigora Mataram, dan teman-teman Laboran dari Lab Matematika UIN Mataram yang telah banyak membantu terbitnya buku ajar ini. Semoga buku ajar ini bermanfaat guna melahirkan mahasiswa yang kompeten dalam bidang bahasa pemrograman khususnya Matlab.

Akhir kata, mengingat setiap ciptaan manusia tidak luput dari kekurangan, maka tim penulis mengucapkan terima kasih dan terbuka untuk menerima saran dari pembaca guna melengkapi atau merevisi buku-buku ajar selanjutnya. Semoga buku ajar ini bermanfaat.

Mataram, 07 Oktober 2018

Tim Penulis



DAFTAR ISI

KATA PENGANTAR.....	v
DAFTAR ISI	vii

BAB 1

KONSTRUKSI SCRIBS PNL.....	1
Tujuan Perkuliahan:	1
1.1 Rasional	1
1.2 Struktur Bahasa Matlab	2
1.3 Scribs Sederhana	3
1.4 PNL Metode Tertutup.....	6
A. Metode Bisection.....	6
B. Metode Regula Falsi	15
1.5 PNL Metode Terbuka.....	22
A. Metode Secant	22
B. Metode Newton Raphson.....	25
C. Metode Fixet Point.....	28
Uji Kompetensi 1.....	31

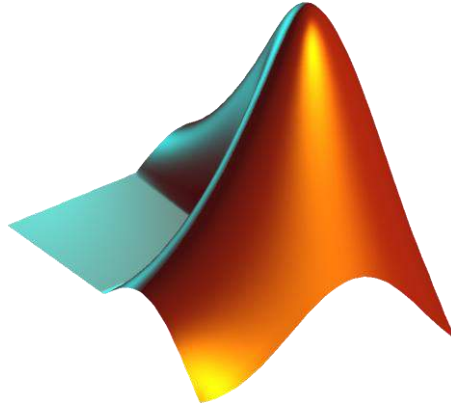
BAB 2

GUI MATLAB PNL & SPNL.....	33
Tujuan Perkuliahan:	33
2.1 Pengantar GUI.....	33
2.2 GUI Sederhana	37
2.3 PNL Metode Tertutup.....	42
2.4 PNL Metode Terbuka.....	50
2.5 Sistem Persamaan Non Linier.....	60
Uji Kompetensi 2.....	67



BAB 3	
RANCANG BANGUN APLIKASI PROJECT	69
Tujuan Perkuliahan:	69
3.1 Membuat <i>MainMenu</i> (Menu Utama)	69
3.2 Membuat Aplikasi Project	73
Daftar Pustaka	79
Glosarium.....	81
Indeks.....	83
Biodata Penulis.....	85





BAB 1

KONSTRUKSI SCRIBS PNL

Tujuan Perkuliahan:

1. Mahasiswa mampu memahami macam-macam struktur bahasa pemrograman di Matlab.
2. Mahasiswa mampu merancang algoritma dan *scribs* pemrograman berbasis *function* dalam menyelesaikan masalah metode numerik

1.1	Rasional
-----	----------

Matematika merupakan alat ilmu pengetahuan untuk menyelesaikan permasalahan atau persoalan dalam kehidupan sehari-hari. Dalam hal tersebut, umumnya Matematika memiliki tiga cara penyelesaian, yaitu metode analitik, numerik dan simulasi. **Metode analitik** merupakan cara penyelesaian permasalahan Matematika dengan menguraikan atau menjabarkan variabel-variabelnya pada model Matematika tersebut. Sebagai contoh misalnya menentukan berapa nilai x yang memenuhi agar $f(x) = x^2 - 2x - 8 = 0$.



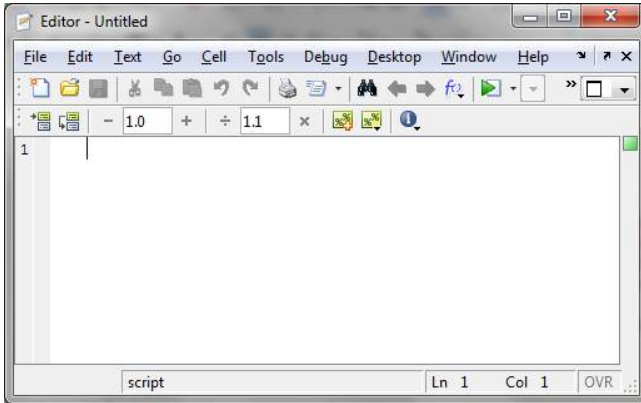
Metode analitik yang dapat digunakan adalah dengan memfaktorkan seperti $f(x)=0 \rightarrow x^2 - 2x - 8 = 0 \rightarrow (x-2)(x-4) = 0$ pernyataan $(x-2)(x-4) = 0$ bernilai benar jika $x = -2$ atau -4 . Oleh karena itu solusi agar $f(x)$ bernilai 0 di berikan oleh $x = -2$ dan $x = -4$ merupakan solusi analitik dari permasalahan tersebut di atas, dan nilai yang diperoleh tersebut bersifat eksak, artinya bahwa solusi yang diperoleh merupakan penyelesaian yang sebenarnya (real) tanpa ada kesalahan. Metode analitik tersebut umumnya hanya dapat menyelesaikan masalah yang sederhana, karena sangat terbatas pada model Matematika yang dimiliki dan kemampuan kecepatan berfikir manusia. Sebagai contoh misalnya berapakah nilai x yang memenuhi agar $f(x) = e^x + 1 = 0$. Tentu saja cara analitik dalam hal ini sulit ditepatkan, oleh karena itu, terdapat banyak metode yang digunakan untuk mendapatkan solusi dari persamaan non linier tersebut. Lima metode yang paling populer adalah metode *bisection*, *regula false*, *secant*, *newton raphson*, dan *Fixet Point*. Masing-masing metode memiliki kelebihan dan kekurangan. Berikut akan dibahas hanya metode terbuka saja, karena metode tertutup sudah dijelaskan di pertemuan sebelumnya.

1.2	Struktur Bahasa Matlab
------------	-------------------------------

M-file atau sering disebut editor merupakan komponen kedua dari Matlab setelah Anda bermain di *Command Windows*. Di *Command Windows* Anda menulis *scribs* langsung *output*, tapi di editor Anda menulis *scribs* kemudian *input* dan *output* akan ditampilkan di *Command Windows*. Belajar menulis *scribs* di editor akan mempermudah dalam proses pembuatan *guide* pada bagian yang akan Anda pelajari berikutnya.

Jika Anda ingin membuat sebuah *m-file*. Anda tinggal ketik kata edit di *Command Windows*, kemudian di-**enter**, maka akan muncul layar seperti berikut.

>> edit



Gambar 1.1 Editor kosong.

Di editor, Anda akan membuat *scribs* (sintaksis) dari sebuah program aplikasi. Nah, Anda akan bertemu dengan kasus atau permasalahan yang berkaitan dengan 3 struktur bahasa, yakni:

1. Struktur urut, yaitu menyelesaikan masalah matematika dengan *scribs* terurut, tanpa ada kontrol.
2. Struktur percabangan, yaitu struktur bahasa dengan adanya kontrol seperti *if ... elseif ... else*, dan *switch ... case of*.
3. Struktur perulangan, yaitu struktur bahasa dengan adanya kontrol seperti *for* dan *while*.

Ketiga struktur ini terkadang terpisah penggunaannya, terkadang juga digunakan secara bersamaan dalam membangun sebuah *scribs*.

1.3 Scribs Sederhana

Struktur urut merupakan struktur paling mudah dalam bahasa pemrograman. Karena tidak adanya *control* yang harus dilakukan pada penulisan *scribs* (sintaksis). Pada dasarnya, sebuah *scribs* terdiri dari beberapa langkah, yakni:



1. Definiskan variable inputan
2. Menulis perintah/rumus
3. Menulis *output*

Sebagai latihan pengantar Anda akan membuat sebuah *scribs* sederhana dengan struktur urut yakni menghitung luas dan keliling persegi panjang. Silahkan Anda buka editor.

Perintah matematika	Perintah Matlab
Membersihkan <i>Command Windows</i>	<code>clear all;</code> <code>clc;</code>
Menulis identitas/judul <i>scribs</i> <code>disp = display =</code> menunjukkan tulisan di <i>Command Windows</i> .	<code>disp('Menghitung Luas dan Keliling Persegi Panjang');</code> <code>disp('Programmer: Syaharuddin');</code>
Input nilai panjang dan keliling	<code>p=input('Panjang = ');</code> <code>l=input('Lebar = ');</code>
Menulis rumus	<code>Luas=p*l;</code> <code>Keliling=2*(p+l);</code>
Menulis <i>output</i>	<code>disp(['Luas = ',num2str(Luas)]);</code> <code>disp(['Keliling = ',num2str(Keliling)]);</code>

Adapun sintaksis keseluruhan seperti berikut.


```

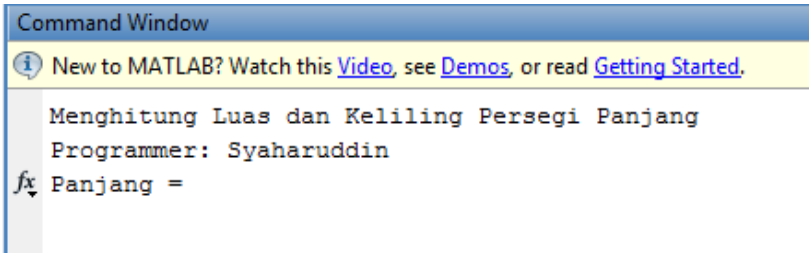
1 - clear all;
2 - clc;
3 - disp('Menghitung Luas dan Keliling Persegi Panjang');
4 - disp('Programmer: Syaharuddin');
5 - p=input('Panjang = ');
6 - l=input('Lebar = ');
7 - Luas=p*l;
8 - Keliling=2*(p+l);
9 - disp(['Luas = ',num2str(Luas)]);
10 - disp(['Keliling = ',num2str(Keliling)]);

```

Gambar 1.2 Tampilan *Scribs* Persegi Panjang

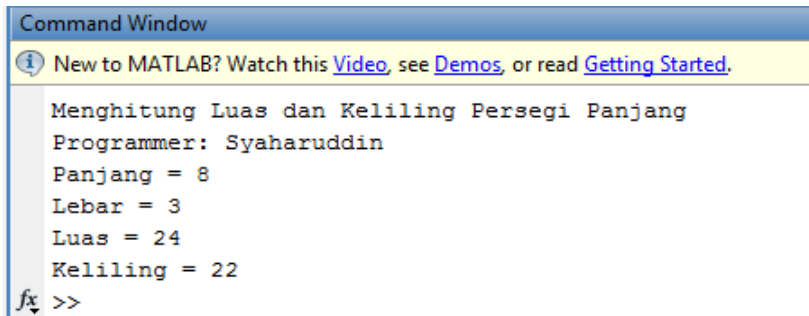


Setelah selesai Anda ketik *scribs*, silahkan tekan tombol eksekusi atau *running* yakni  atau bisa juga tekan **F5** di keyboard. Proses ini untuk menyimpan *scribs* Anda terlebih dahulu. Buatlah folder baru dengan nama “**Kumpulan M-file**”, kemudian simpalah *m-file* Anda dengan nama “**Persegi_Panjang**” (ingat!, tidak boleh spasi), kemudian klik OK, maka akan muncul seperti berikut.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Menghitung Luas dan Keliling Persegi Panjang
Programmer: Syaharuddin
fx Panjang =
```

Silahkan Anda *input* nilai panjang, kemudian enter *input* lagi nilai lebar, kemudian enter lagi, maka akan keluar hasil luas dan keliling, seperti berikut.



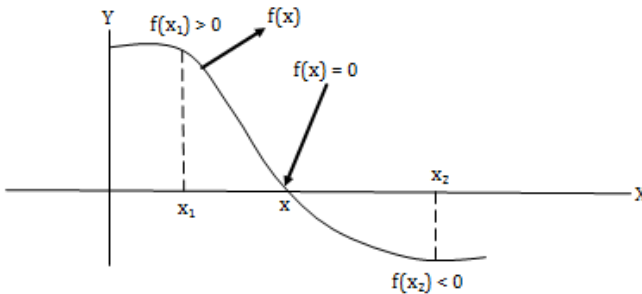
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Menghitung Luas dan Keliling Persegi Panjang
Programmer: Syaharuddin
Panjang = 8
Lebar = 3
Luas = 24
Keliling = 22
fx >>
```

Jika Anda ingin menginput nilai yang baru, silahkan ketik kembali nama *m-file* tadi, yakni “**Persegi_Panjang**”, kemudian enter. Jika lupa nama *m-file*, Anda bisa ke *editor* lagi kemudian klik tombol *running* (F5).

A. Metode Bisection

Bisection terdiri dari dua kata, yakni *bi* = dua dan *section* = bagian, sehingga metode *bisection* merupakan cara penyelesaian persamaan non linier dengan membuat dua buah bagian interval dari domain penyelesaian persamaan non linier tersebut. Proses pembagian interval tersebut mula-mula diawali dengan penentuan interval yang memuat solusi (akar) untuk $f(x)$. Berikut diberikan ilustrasi grafis metode *Bisection*.

1. Mula-mula pilih x_1 dan x_2 sembarang sehingga $f(x_1)f(x_2) < 0$



Gambar 1.3 Interval yang memuat x sehingga $f(x) = 0$

Pada tahap ini, prinsipnya adalah menentukan lokasi titik potong $f(x)$ dengan sumbu x . Kondisi ini terjadi apabila x_1 dan x_2 yang dipilih memberikan nilai $f(x_1)$ dan $f(x_2)$ berlainan tanda (positif atau negatif) sebagaimana pada gambar 1.2 diatas dapat ditentukan dengan jika $f(x_1)f(x_2) < 0$, maka diantara ada yang negatif dan yang lainnya positif.

2. Hitung nilai $x_r = 0.5(x_1 + x_2)$

Pada tahap ini sesungguhnya dilakukan upaya membagi dua interval akar menjadi dua buah bagian yang sama. Titik baginya disebut x_r sehingga bagian-bagian intervalnya menjadi $x_1 < x < x_r$ dan $x_1 < x < x_2$.



3. Periksa nilai x_r . Posisi nilai x_r ada tiga kemungkinan, yakni (1) sebelah kiri akar, (2) sebelah kanan akar atau (3) tepat pada titik akar.
4. Perbaharui interval akar persamaan sebelumnya

Pada tahap ini dilakukan pemilihan bagian interval yang memuat akar persamaan. Mengacu kepada gambar 1.5 di atas, maka dapat dikontuksikan tolak ukur interval yang baru sebagai berikut.

a. Berdasarkan nilai $f(x_1)$

✓ Jika $f(x_1)f(x_r) > 0$, maka $x_1 = x_r$ (geser posisi x_1 ke x_r)

Diperoleh Interval baru $[x_1 x_2] = [x_r x_2]$

✓ Jika $f(x_1)f(x_r) < 0$, maka $x_1 = x_r$ (geser posisi x_1 ke x_r)

Diperoleh Interval baru $[x_1 x_2] = [x_1 x_r]$

✓ $f(x_1)f(x_r) = 0$, maka akar = x_r .

Interval tidak perlu di buat, karena x_r adalah nilai x sehingga $f(x) = 0$.

b. Berdasarkan $f(x_2)$

✓ Jika $f(x_1)f(x_r) < 0$, maka $x_1 = x_r$ (geser posisi x_1 ke x_r)

Diperoleh Interval baru $[x_1 x_2] = [x_r x_2]$

✓ Jika $f(x_1)f(x_r) > 0$, maka $x_1 = x_r$ (geser posisi x_1 ke x_r)

Diperoleh Interval baru $[x_1 x_2] = [x_1 x_r]$

✓ $f(x_1)f(x_r) = 0$, maka akar = x_r .

Interval tidak perlu di buat, karena x_r adalah nilai x sehingga $f(x) = 0$.

5. Kembali lagi diulangi membagi interval akar yang baru diperoleh dengan mengikuti langkah 2, 3 dan 4 di atas sehingga diperoleh nilai $f(x_r) = 0$ atau $f(x_r)$ sedekat mungkin dengan 0 (nol).

Mencermati uraian proses metode *Bisection* di atas, maka dapat dirumuskan suatu algoritma program sebagai berikut:



Algoritma Program Metode Bisection

Step 0. Mulai

Step 1. Tentukan interval $[x_1, x_2]$

Step 2. Hitung nilai $f(x_1)$ dan $f(x_2)$

Step 3. Jika $f(x_1)f(x_2) < 0$, maka kerjakan step 4 sampai 11

Step 4. Masukkan toleransi error (E)

Step 5. Ulangin terus step 6 sampai 11 jika $|f(x_r)| > e$

Step 6. Hitung nilai $x_r = 0.5(x_1 + x_2)$

Step 7. Hitung nilai $f(x_r)$

Step 8. Jika $|f(x_r)| > e$, maka kerjakan step 9 sampai 11

Step nilai $f(x_1)$

Step 10. Jika $f(x_1)f(x_2) < 0$, maka $x_2 = x_r$

Step 11. Jika $f(x_1)f(x_2) > 0$, maka $x_1 = x_r$

Step 12. Jika $f(x_1)f(x_2) > 0$, maka tidak ada akar pada $[x_1, x_2]$

Step 13. selesai

Contoh 1:

Tentukan nilai x dengan menggunakan metode Bisection $xe^{-x} + 1 = 0$ dengan toleransi kesalahan $E = 0.001$.

Solusi:

Dengan mengembangkan bahasa program standar pada algoritma di atas sebagai berikut:

$$f(x) = x * \exp(-x) + 1$$

$$[x_1, x_2] = [-1, 0]$$

$$f(-1) = -1.7183$$

$$f(0) = 1$$

Karena $f(x_1)f(x_2) < 0$, maka x sehingga $f(x) = 0$ pada interval $[-1, 0]$

Toleransi kesalahan = 0.001



Iterasi ke-1

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-1 + 0) \\ &= 0.5(-1) \\ &= -0.5\end{aligned}$$

$$f(x_r) = 0.17564$$

Karena $|f(x_r)| = 0.17564 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_1 = x_r$

Jadi interval baru adalah $[-1 - 0.5]$

Iterasi ke-2

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-1 + 0.5) \\ &= 0.5(-1.5) \\ &= 0.75\end{aligned}$$

$$f(x_r) = -0.58775$$

Karena $|f(x_r)| = 0.58775 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) > 0$, sehingga $x_1 = x_r$

Karena $|f(x_r)| = 0.58775 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) > 0$, sehingga $x_1 = x_r$

Jadi interval baru adalah $[-0.75 - 0.5]$

Jadi interval baru adalah $[-0.75 - 0.5]$

Iterasi ke-3

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.75 + 0.5) \\ &= 0.5(-1.5) \\ &= -0.75\end{aligned}$$

$$f(x_r) = -0.58775$$

Iterasi ke-4

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.75 + -0.5) \\ &= 0.5(-1.25) \\ &= -0.625\end{aligned}$$

$$f(x_r) = -0.16765$$



Karena $|f(x_r)| = 0.16765 > 0.001 = e$ maka *update* $[x_1 \ x_2]$
Karena $f(x_1) = -0.58775$ maka $f(x_1)f(x_r) > 0$, sehingga $x_1 = x_r$
Jadi interval baru adalah $[-0.625 \ -0.5]$

Iterasi ke-5

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.625 + -0.5) \\ &= 0.5(-1.125) \\ &= -0.5625\end{aligned}$$

$$f(x_r) = 0.012782$$

Karena $|f(x_r)| = 0.012782 > 0.001 = e$ maka *update* $[x_1 \ x_2]$
Karena $f(x_1) = -0.16765$ maka $f(x_1)f(x_r) < 0$, sehingga $x_1 = x_r$
Jadi interval baru adalah $[-0.625 \ -0.5625]$

Iterasi ke-6

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.625 + -0.5625) \\ &= 0.5(-1.1875) \\ &= -0.59375\end{aligned}$$

$$f(x_r) = -0.075142$$

Karena $|f(x_r)| = 0.075142 > 0.001 = e$ maka *update* $[x_1 \ x_2]$
Karena $f(x_1) = -0.16765$ maka $f(x_1)f(x_r) > 0$, sehingga $x_1 = x_r$
Jadi interval baru adalah $[-0.59375 \ -0.5625]$

Iterasi ke-7

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.59375 + -0.5625) \\ &= 0.5(-1.1563) \\ &= -0.57813\end{aligned}$$

$$f(x_r) = -0.030619$$

Karena $|f(x_r)| = 0.030619 > 0.001 = e$ maka *update* $[x_1 \ x_2]$
Karena $f(x_1) = -0.075142$ maka $f(x_1)f(x_r) > 0$, sehingga $x_1 = x_r$
Jadi interval baru adalah $[-0.57813 \ -0.5625]$



Iterasi ke-8

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.57813 + -0.5625) \\ &= 0.5(-1.1406) \\ &= -0.57031\end{aligned}$$

$$f(x_r) = -0.00878$$

Karena $|f(x_r)| = 0.00878 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -0.030619$ maka $f(x_1)f(x_r) > 0$, sehingga $x_1 = x_r$

Jadi interval baru adalah $[-0.57031 \ -0.5625]$

Iterasi ke-9

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.57031 + -0.5625) \\ &= 0.5(-1.1328) \\ &= -0.56641\end{aligned}$$

$$f(x_r) = -0.0020354$$

Karena $|f(x_r)| = 0.0020354 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -0.00878$ maka $f(x_1)f(x_r) < 0$. Sehingga $x_1 = x_r$

Jadi interval baru adalah $[-0.57031 \ -0.56641]$

Iterasi ke-10

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.57031 + -0.56641) \\ &= 0.5(-1.1367) \\ &= -0.56836\end{aligned}$$

$$f(x_r) = -0.0033637$$

Karena $|f(x_r)| = 0.0033637 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -0.00878$ maka $f(x_1)f(x_r) < 0$. Sehingga $x_1 = x_r$

Jadi interval baru adalah $[-0.56836 \ -0.56641]$

Iterasi ke-10

$$\begin{aligned}x_r &= 0.5(x_1 + x_2) \\ &= 0.5(-0.56836 + -0.56641) \\ &= 0.5(-1.1348) \\ &= -0.56738\end{aligned}$$

$$f(x_r) = -0.00066198$$



Karena $|f(x_r)| = 0.00066198 > 0.001 = e$ maka proses berhenti
 Jadi akar persamaan adalah $x = -0.56738$ dengan $f(x_r) = -0.00066198$

Dari hasil perhitungan di atas, untuk mendapatkan nilai x sehingga $f(x)$ dekat dengan 0.001 di butuhkan 10 iterasi (proses pengulangan perhitungan) dengan nilai $x = -0.56738$ dan $f(x) = -0.00066190$. grafik konvergensi error atau selisih antara $f(x)$ dengan sumbu x untuk tiap iterasi serta grafik pendekatan solusi akar persamaan $f(x)$.

Adapun konstruksi *scribs* atau *m-function* Metode *Bisection* sebagai berikut:

```
function [H] = bisection(f,x1,x2,e)
syms x;
H=[{'f(x)=',char(f)}];
H=[H;{'[x1 x2] = ',num2str(x1), ' ',num2str(x2), '']};
H=[H;{'e =',num2str(e)}];
f1=(eval(subs(f,x1)));
f2=(eval(subs(f,x2)));
H=[H;{'f(',num2str(x1),') = ',num2str(f1)}];
H=[H;{'f(',num2str(x2),') = ',num2str(f2)}];
if f1*f2<0;
    H=[H;{'Karena f(x1)f(x2)<0, maka x shg f(x)= 0 pada
interval [' ,...
        num2str(x1), ' ',num2str(x2), '']}]};
    fr=1;c=0;
    while abs(fr)>e;
        c=c+1;
        H=[H;{' '};{'Iterasi ke-',num2str(c)}];
        xr=0.5*(x1+x2);
        fr=(eval(subs(f,xr)));
        H=[H;{' xr = 0.5(x1 + x2)'}];
        H=[H;{'[ = 0.5(',num2str(x1), ' + ',num2str(x2), ')]'}];
        H=[H;{'[ = 0.5(',num2str(x1+x2), ')]'}];
        H=[H;{'[ = ',num2str(xr)]'}];
        H=[H;{'f(xr)= ',num2str(fr)}];
        E(c)=abs(fr);
        if abs(fr)>e;
            H=[H;{'Karena |f(xr)| =
',num2str(abs(fr)), '>',num2str(e), '...
            'e maka update [x1 x2]'}];
            f1=(eval(subs(f,x1)));
            if fr*f1<0;
                x2=xr;
                h1=[{'Karena f(x1) = ',num2str(f1), ' maka
f(x1)f(xr)<0, shg x2=xr'}];
            end
        end
    end
end
```



```

        h2=[{(['Jadi interval baru adalah
[' ,num2str(x1), ...
        ' ',num2str(x2), '']')]];
    else
        x1=xr;
        h1=[{(['Karena f(x1) = ',num2str(f1), ' maka
f(x1)f(xr)>0, shg x1=xr']')]];
        h2=[{(['Jadi interval baru adalah
[' ,num2str(x1), ...
        ' ',num2str(x2), '']')]];
    end
    H=[H;h1;h2];
else
    H=[H;{(['Karena |f(xr)| =
',num2str(abs(fr)), '<',num2str(e), ...
    'e maka proses berhenti']')});
end
end
H=[H;{ ' ' };{(['Jadi akar persamaan adalah x =
',num2str(xr), ...
    ' dengan f(xr) = ',num2str(fr)]')});
subplot(2,1,1)
plot(E, '-*r')
grid on
xlabel('Iterasi')
ylabel('Jarak f(x) dengan 0 ')
title('Grafik konvergensi error Bisection')
subplot(2,1,2)
plot(xr, fr, '*r')
hold on
ezplot(f, [xr-3 xr+3])
grid on
legend([' x = ',num2str(xr)], ['f(xr)= ',num2str(fr)])
else
    H=(['Karena f(x1)f(x2)>0, maka tidak ada x shg f(x)=0 pada
interval [' , ...
    num2str(x1), ' ',num2str(x2), '']')]);
end
end
end

```

Contoh Simulasi:

```

>> syms x
>> f=2*x*exp(-x+1)+2
f =
2*x*exp(1 - x) + 2
>> x1=-0.28125;
>> x2=-0.25
x2 =
-0.2500

```



```

>> e=0.001
e =
    1.0000e-03
>> bisection(f,-0.28125,-0.25,0.001)
ans =

'f(x)=2*x*exp(1 - x) + 2'
'[x1 x2] = [-0.28125 -0.25]'
'e =0.001'
'f(-0.28125) = -0.02564'
'f(-0.25) = 0.25483'
'Karena f(x1)f(x2)<0, maka x shg f(x)= 0 pada interval [-
0.28125 -0.25]' ' '
'Iterasi ke-1'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.28125 + -0.25) '
' = 0.5(-0.53125) '
' = -0.26563 '
'f(xr)= 0.11656 '
'Karena |f(xr)| = 0.11656>0.001=e maka update [x1 x2]'
'Karena f(x1) = -0.02564 maka f(x1)f(xr)<0, shg x2=xr'
'Jadi interval baru adalah [-0.28125 -0.26563]'
' '
'Iterasi ke-2'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.28125 + -0.26563) '
' = 0.5(-0.54688) '
' = -0.27344 '
'f(xr)= 0.045953 '
'Karena |f(xr)| = 0.045953>0.001=e maka update [x1 x2]'
'Karena f(x1) = -0.02564 maka f(x1)f(xr)<0, shg x2=xr'
'Jadi interval baru adalah [-0.28125 -0.27344]'
' '
'Iterasi ke-3'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.28125 + -0.27344) '
' = 0.5(-0.55469) '
' = -0.27734 '
'f(xr)= 0.010281 '
'Karena |f(xr)| = 0.010281>0.001=e maka update [x1 x2]'
'Karena f(x1) = -0.02564 maka f(x1)f(xr)<0, shg x2=xr'
'Jadi interval baru adalah [-0.28125 -0.27734]'
' '
'Iterasi ke-4'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.28125 + -0.27734) '
' = 0.5(-0.55859) '
' = -0.2793 '
'f(xr)= -0.0076483 '
'Karena |f(xr)| = 0.0076483>0.001=e maka update [x1 x2]'

```



```

'Karena f(x1) = -0.02564 maka f(x1)f(xr)>0, shg x1=xr'
'Jadi interval baru adalah [-0.2793 -0.27734]'
' '
'Iterasi ke-5'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.2793 + -0.27734) '
' = 0.5(-0.55664) '
' = -0.27832 '
'f(xr)= 0.0013242'
'Karena |f(xr)| = 0.0013242>0.001=e maka update [x1 x2]'
'Karena f(x1) = -0.0076483 maka f(x1)f(xr)<0, shg x2=xr'
'Jadi interval baru adalah [-0.2793 -0.27832]'
' '
'Iterasi ke-6'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.2793 + -0.27832) '
' = 0.5(-0.55762) '
' = -0.27881 '
'f(xr)= -0.0031601'
'Karena |f(xr)| = 0.0031601>0.001=e maka update [x1 x2]'
'Karena f(x1) = -0.0076483 maka f(x1)f(xr)>0, shg x1=xr'
'Jadi interval baru adalah [-0.27881 -0.27832]'
' '
'Iterasi ke-7'
' xr = 0.5(x1 + x2) '
' = 0.5(-0.27881 + -0.27832) '
' = 0.5(-0.55713) '
' = -0.27856 '
'f(xr)= -0.00091748'
'Karena |f(xr)| = 0.00091748<0.001=e maka proses berhenti'
' '
'Jadi akar persamaan adalah x = -0.27856 dengan f(xr) = -
0.00091748'

```

B. Metode Regula Falsi

Metode *Regula Falsi* merupakan cara untuk menentukan akar persamaan suatu fungsi, dengan melakukan pengulangan akar palsu yang dibentuk dari perpotongan garis yang melalui titik $(x_1, f(x_1))$ dan $(x_2, f(x_2))$ dengan sumbu x , ilustrasi grafis sebagai berikut:

Persamaan garis yang melalui titik $(x_1, f(x_1))$ dan $(x_2, f(x_2))$ adalah

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - f(x_1)}{f(x_2) - f(x_1)} \Rightarrow x = \frac{(x_2 - x_1)(y - f(x_1))}{f(x_2) - f(x_1)} + x_1$$



Jika $x = x_r$, maka diperoleh $y = 0$, kenyataan ini memberikan persamaan regula falsi sebagai berikut:

$$x_r = x_1 - \frac{f(x_1)(x_2 - x_1)}{f(x_2) - f(x_1)}$$

Selanjutnya tahapan-tahapan pengerjaan penentuan akar persamaan $f(x)$ dengan metode betode *Regula Falsi* sama dengan tahapan-tahapan sebagaimana metode *Bisection*. Berdasarkan teori di atas, maka algoritma program metode *Regula Falsi* dapat dirumuskan sebagai berikut:

Algoritma Program Metode Regula Falsi

Step 0. Mulai

Step 1. Tentukan interval $[x_1 \ x_2]$

Step 2. Hitung nilai $f(x_1)$ dan $f(x_2)$

Step 3. Jika $f(x_1)f(x_2) < 0$, maka kerjakan step 4 sampai 11

Step 4. Masukkan toleransi error (E)

Step 5. Ulangi terus step 6 sampai 11 jika $|f(x_r)| > e$

Step 6. Hitung nilai $x_r = x_1 - \frac{x_2 f(x_1) x_1 f(x_1)}{f(x_2) - f(x_1)}$

Step 7. Hitung nilai $f(x_r)$

Step 8. Jika $|f(x_r)| > e$, maka kerjakan step 9

Step 9. Hitung nilai $f(x_1)$

Step 10. Jika $f(x_1) f(x_r) < 0$, maka $x_2 = x_r$

Step 11. Jika $f(x_1) f(x_r) < 0$, maka $x_1 = x_r$

Step 12. jika $f(x_1) f(x_2) > 0$, maka tidak ada akar pada $[x_1 x_2]$

Step 13. Selesai

Contoh 2:

Tentukan nilai x dengan menggunakan metode *regula falsi* sehingga $x e^{-x} + 1 = 0$ dengan toleransi kesalahan $E = 0.001$



Solusi:

Penyelesaian dengan program computer setelah mengkonversi algoritma komputasi, maka diperoleh *output* sebagai berikut:

$$f(x) = x \cdot \exp(-x) + 1$$

$$[x_1 \ x_2] = [-1 \ 0]$$

$$f(-1) = -1.7183$$

$$f(0) = 1$$

Karena $f(x_1)f(x_2) < 0$, maka x sehingga $f(x) = 0$ pada interval $[-1 \ 0]$

Toleransi kesalahan = 0.001

Iterasi ke-1

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\&= -1 - (0 - (-1))f(-1)/(f(0) - f(-1)) \\&= -1 - (1)(-1.7183)/(1 - -1.7183) \\&= -1 - 1.7183/(2.7183) \\&= -1 - (-0.63212) \\&= -0.36788\end{aligned}$$

$$f(x_r) = 0.46854$$

Karena $|f(x_r)| = 0.46854 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_2 = x_r$

Jadi interval baru adalah $[-1 \ -0.36788]$

Iterasi ke-2

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\&= -1 - (-0.36788 - (-1))f(-1)/(f(-0.36788) - f(-1)) \\&= -1 - (0.63212)(-1.7183)/(0.46854 - -1.7183) \\&= -1 - (-1.0862)/(2.1868) \\&= -1 - (-0.49669) \\&= -0.50331\end{aligned}$$

$$f(x_r) = 0.16472$$

Karena $|f(x_r)| = 0.1674 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_2 = x_r$

Jadi interval baru adalah $[-1 \ -0.50331]$



Iterasi ke-3

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\&= -1 - (-0.50331 - -1)f(-1)/(f(-0.50331)) - f(-1) \\&= -1 - (0.49669)(-1.7183)/(0.16742 - -1.7183) \\&= -1 - 0.85345/(1.8857) \\&= -1 - 0.45259 \\&= -0.54741\end{aligned}$$

$$f(x_r) = 0.053649$$

Karena $|f(x_r)| = 0.053649 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_2 = x_r$

Jadi interval baru adalah $[-1 \ -0.54741]$

Iterasi ke-4

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\&= -1 - (-0.54741 - -1)f(-1)/(f(-0.54741)) - f(-1) \\&= -1 - (0.45259)(-1.7183)/(0.053649 - -1.7183) \\&= -1 - (-0.7767)/(1.7719) \\&= -1 - (0.43888) \\&= -0.56112\end{aligned}$$

$$f(x_r) = 0.016575$$

Karena $|f(x_r)| = 0.016575 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_2 = x_r$

Jadi interval baru adalah $[-1 \ -0.56112]$

Iterasi ke-5

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\&= -1 - (-0.56112 - -1)f(-1)/(f(-0.56112)) - f(-1) \\&= -1 - (0.43888)(-1.7183)/(0.016575 - -1.7183) \\&= -1 - (-0.75413)/(1.7349) \\&= -1 - (-0.43469) \\&= -0.56531\end{aligned}$$

$$f(x_r) = 0.0050629$$

Karena $|f(x_r)| = 0.0050629 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_2 = x_r$

Jadi interval baru adalah $[-1 \ -0.56531]$



Iterasi ke-6

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\ &= -1 - (-0.56531 - -1)f(-1)/(f(-0.56531)) - f(-1) \\ &= -1 - (0.43469)(-1.7183)/(0.0050629 - -1.7183) \\ &= -1 - (-0.74692)/(1.7233) \\ &= -1 - (-0.43341) \\ &= -0.56659\end{aligned}$$

$$f(x_r) = 0.001541$$

Karena $|f(x_r)| = 0.001541 > 0.001 = e$ maka *update* $[x_1 \ x_2]$

Karena $f(x_1) = -1.7183$ maka $f(x_1)f(x_r) < 0$, sehingga $x_2 = x_r$

Jadi interval baru adalah $[-1 \ -0.56659]$

Iterasi ke-7

$$\begin{aligned}x_r &= x_1 - (x_2 - x_1)f(x_1)/(f(x_2) - f(x_1)) \\ &= -1 - (-0.56659 - -1)f(-1)/(-0.56659) - f(-1) \\ &= -1 - (0.43341)(-1.7183)/(0.001541 - -1.7183) \\ &= -1 - (-0.74473)/(1.7198) \\ &= -1 - (-0.43303) \\ &= -0.56697\end{aligned}$$

$$f(x_r) = 0.00046855$$

Karena $|f(x_r)| = 0.00046855 > 0.001 = e$ maka proses berhenti

Jadi akar persamaan adalah $x = 0.56697$ dengan $f(x_r) = 0.00046855$

Hasil komutasi di atas, menunjukkan bahwa solusi pendekatan untuk x sehingga $f(x) = 0$ adalah $x = 0.56697$. nilai tersebut diperoleh setelah melakukan perhitungan hingga iterasi ke-7. Konvergensi iterasi ini lebih cepat jika dibandingkan dengan Metode Bisection.

Konstruksi Scribs M-Function Metode Regula Falsi

```
function [H] =RF(f, x1, x2, e)
syms x;
H=[{'f(x)=' , char(f)}];
H=[H; {'[x1 x2] = [' , num2str(x1) , ' ' , num2str(x2) , ' ]'}];
H=[H; {'e = ' , num2str(e)}];
f1=eval(subs(f, x1));
f2=eval(subs(f, x2));
H=[H; {'f(' , num2str(x1) , ') = ' , num2str(f1)}];
```



```

H=[H;{(['f(',num2str(x2),'') = ',num2str(f2))}]);
if f1*f2<0
    H=[H;{(['Karena f(x1)f(x2)<0, maka x shg f(x)= 0 pada
interval [' ,...
        num2str(x1),' ',num2str(x2),'']')});
    fr=1;c=0;
    while abs(fr)>e;
        c=c+1;
        H=[H;{(['Iterasi ke-',num2str(c))}]);
        f1=eval(subs(f,x1));f2=eval(subs(f,x2));
        xr=x1-((x2*f1-x1*f2)/(f2-f1));
        fr=eval(subs(f,xr));
        H=[H;{(['xr = x1 - (x2-x1)f(x1)/(f(x2)-f(x1))']')});
        H=[H;{([' = ',num2str(x1),'-',num2str(x2),' -
',num2str(x1),...
            ' )f(',num2str(x1),'')/(f(',num2str(x2),'')-
f(',num2str(x1),'')')')});
        H=[H;{([' = ',num2str(x1),'-',num2str(x2-x1),...
            ' )(',num2str(f1),'')/(',num2str(f2),'
            ' )(',num2str(f1),'')')});
        H=[H;{([' = ',num2str(x1),'-',num2str((x2-
x1)*f1),'')/(',num2str(f2-f1),'')')});
        H=[H;{([' = ',num2str(x1),'-',num2str((x2-x1)*f1/(f2-
f1)),'')')});
        H=[H;{([' = ',num2str(xr))}]);
        H=[H;{(['f(xr) = ',num2str(fr))}]);
        E(c)=abs(fr);
        if abs(fr)>e;
            H=[H;{(['Karena |f(xr)| =
',num2str(abs(fr)),'>',num2str(e),...
                'e maka update [x1 x2]']')});
            f1=eval(subs(f,x1));
            if fr*f1<0;
                x2=xr;
                h1=[(['Karena f(x1) = ',num2str(f1),' maka
f(x1)f(xr)<0, shg x2=xr']')});
                h2=[(['Jadi interval baru adalah
',num2str(x1),...
                    ' ',num2str(x2),'']')});
            else
                x1=xr;
                h1=[(['Karena f(x1) = ',num2str(f1),' maka
f(x1)f(xr)>0, shg x1=xr']')});
                h2=[(['Jadi interval baru adalah
',num2str(x1),...
                    ' ',num2str(x2),'']')});
            end
            H=[H;h1;h2];
        else
            H=[H;{(['Karena |f(xr)| =

```



```

    ',num2str(abs(fr)), '<', num2str(e), ...
        '=e maka proses berhenti']]);
    end
end
H=[H;{(['Jadi akar persamaan adalah x = ', num2str(xr), ...
    ' dengan f(xr) = ', num2str(fr)])}];
subplot(2,1,1)
plot(E, '--*r')
grid on
xlabel('Iterasi')
ylabel('Jarak f(x) dengan 0 ')
title('Grafik konvergensi error Regula_Falsi')
subplot(2,1,2)
plot(xr, fr, '*r')
hold on
ezplot(f, [xr-3 xr+3])
grid on
legend([' x = ', num2str(xr)], ['f(xr)= ', num2str(fr)])
else
    H=[(['Karena f(x1)f(x2)>0, maka tidak ada x shg f(x)=0 pada
interval [', ...
    num2str(x1), ' ', num2str(x2), ']' )]);
end
end
end

```

Contoh Simulasi:

```

>> RF(f,-0.28125,-0.25,0.001)
ans =
    'f(x)=2*x*exp(1 - x) + 2'
    '[x1 x2] = [-0.28125 -0.25]'
    'e =0.001'
    'f(-0.28125) = -0.02564'
    'f(-0.25) = 0.25483'
    'Karena f(x1)f(x2)<0, maka x shg f(x)= 0 pada interval [-
0.28125 -0.25]'
    '
    'Iterasi ke-1'
    'xr = x1 - (x2-x1)f(x1)/(fx(2)-f(x1))'
    ' = -0.28125-(-0.25 - -0.28125)f(-0.28125)/(f(-0.25)-f(-
0.28125))'
    ' = -0.28125-(0.03125)(-0.02564)/(0.25483 - -0.02564)'
    ' = -0.28125-(-0.00080126)/(0.28047)'
    ' = -0.28125-(-0.0028569)'
    ' = -0.27839'
    'f(xr)= 0.00065558'
    'Karena |f(xr)| = 0.00065558<0.001=e maka proses berhenti'
'Jadi akar persamaan adalah x = -0.27839 dengan f(xr) =
0.00065558'

```



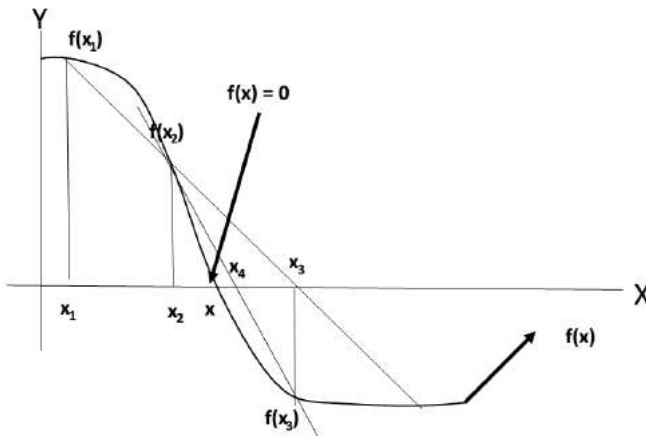
Metode terbuka adalah cara penentuan nilai x sehingga $f(x) = 0$ dengan tidak harus mengurung akar persamaan. Dalam hal ini, interval pencarian atau nilai yang di berikan tidak perlu muat akar persamaan. Tiga jenis metode yang tergolong metode terbuka diantaranya adalah *Metode Secant*, *Metode Newton Raphson*, dan *Fixet Point*.

A. Metode Secant

Metode *secant* merupakan cara untuk mendapatkan nilai x sehingga $f(x) = 0$ yang bersifat terbuka, yakni penentuan nilai x_1 dan x_2 tidak perlu harus mengurung akar persamaan sebagaimana pada metode *bisection* dan *regulasi falsi*.

Adapun Algoritma Metode Secant:

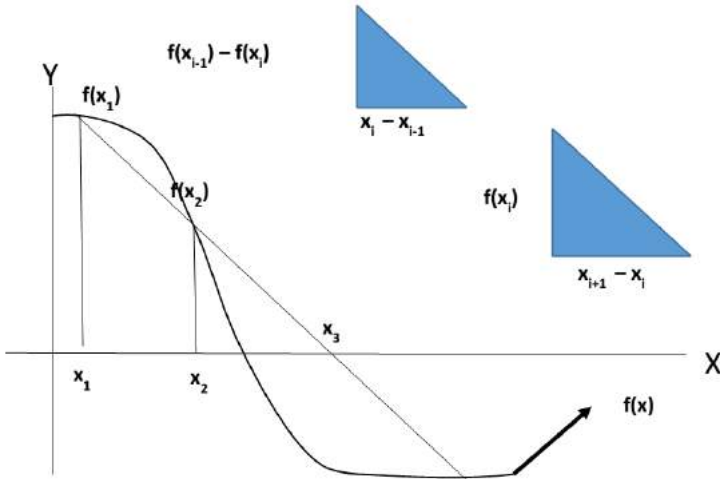
- Pilih sembarang x_1 dan x_2 .
- Melalui titik $(x_1, f(x_1))$ dan $(x_2, f(x_2))$ Tarik garis hingga memotong sumbu x . titik potongnya disebut x_3 .
- Melalui titik $(x_2, f(x_2))$ dan $(x_3, f(x_3))$ Tarik garis hingga memotong sumbu x . titik potongnya disebut x_4 . Gambar 1.1



Gambar 1.4 Model Fungsi Menggunakan Metode Secant



Proses tersebut dilanjutkan hingga diperoleh $f(x_n) = 0$ atau $f(x_n)$ sedekat mungkin dengan 0 (nol).



Gambar 1.5 Model $f(x)$ dan Titik Awal Metode Secant

Dari dua buah segitiga yang dikonstruksi, maka dapat dibuat nilai perbandingan sebagai berikut :

$$\begin{aligned} \frac{x_{i+1} - x_i}{x_i - x_{i-1}} &= \frac{f(x_i)}{f(x_{i-1}) - f(x_i)} \Rightarrow x_{i+1} \\ &= x_i + \frac{(x_i - x_{i-1})f(x_i)}{f(x_{i-1}) - f(x_i)} \\ &\Rightarrow x_{i+1} = x_i - \frac{(x_i - x_{i-1})f(x_i)}{f(x_i) - f(x_{i-1})} \end{aligned}$$

Jadi persamaan iterative dari metode secant adalah

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



Konstruksi *Scribs* M-Function Metode Secant

```

function [H] = secant(f,x1,x2,e)
syms x;
X=[x1 x2];
H=[{'f(x)=',char(f)}];
H=[H;{'[x1 x2] = ',num2str(x1),' ',num2str(x2),' '}]';
H=[H;{'e = ',num2str(e)}];
Fx=eval(subs(f,X));c=2;
while abs(Fx(end))>e;
    H=[H;{' '};{'Iterasi ke-',num2str(c-1)}];
    b=eval(subs(f,X(c)));
    g=eval(subs(f,X(c-1)));
    X(c+1)=X(c)-(X(c)-X(c-1))*b/(b-g);
    H=[H;{'x',num2str(c+1),' = ' , num2str(c),' -
(x',num2str(c),' -x',...
num2str(c-1),' )f(x',num2str(c),' )/f(x',num2str(c),' -
f(x',num2str(c-1),' )')}]';
    H=[H;{' = ',num2str(X(c)), '-(' , num2str(X(c)), ' -
',num2str(X(c-1)), ') (' , num2str(b), ') / (' , num2str(b), ' -
',num2str(g), ')')}]';
    H=[H;{'( ' = ',num2str(X(c)), '-(' , num2str(X(c)-X(c-
1)), ') (' , num2str(b), ') / (' , num2str(b-g), ')')}]';
    H=[H;{'( ' = ',num2str(X(c)), '-(' , num2str((X(c)-X(c-
1))*b)/(b-g), ')')}]';
    H=[H;{' = ',num2str(X(c+1))}]';
    Fx=eval(subs(f,X));
    if abs(Fx(end))>e;
        h=[{'|f(x',num2str(c+1),' )| = ' , ...
|',num2str(Fx(end)), '| = ' , ...
num2str(abs(Fx(end))), ' > ' , num2str(e), ' = e,
maka Lanjut'}]'];
    else
        h=[{'|f(x',num2str(c+1),' )| = ' , ...
|',num2str(Fx(end)), '| = ' , ...
num2str(abs(Fx(end))), ' < ' , num2str(e), ' = e,
maka Stop'}]'];
    end
    c=c+1;
    H=[H;h];
end
H=[H;{' '};{'Jadi akar persamaan adalah x =
',num2str(X(end)), '...
' dengan f(x) = ',num2str(Fx(end))}]'];
subplot(2,1,1);
plot(abs(Fx),'-r');
grid on;
xlabel('Iterasi');
ylabel('Jarak f(x) dengan 0 ');
title('Grafik konvergensi error Secant');
subplot(2,1,2);

```



```

plot(X(end), Fx(end), '*r');
hold on;
ezplot(f, [X(end)-3 X(end)+3]);
grid on;
legend(['          x          =          ', num2str(X(end))], ['f(xr) = ', num2str(Fx(end))]);
end

```

Contoh Simulasi:

```

>> secant(f, -0.28125, -0.25, 0.001)
ans =
'f(x)=2*x*exp(1 - x) + 2'
'[x1 x2] = [-0.28125 -0.25]'
'e =0.001'
''
'Iterasi ke-1'
'x3 = x2-(x2-x1) f(x2)/f(x2)-f(x1)'
' = -0.25-(-0.25 - -0.28125) (0.25483)/(0.25483 - -0.02564)'
' = -0.25-(0.03125) (0.25483)/(0.28047)'
' = -0.25-(0.05689)'
' = -0.27839'
'|f(x3)| = |0.00065558| = 0.00065558 < 0.001 = e, maka Stop'
''
'Jadi akar persamaan adalah x = -0.27839 dengan f(x) = 0.00065558'

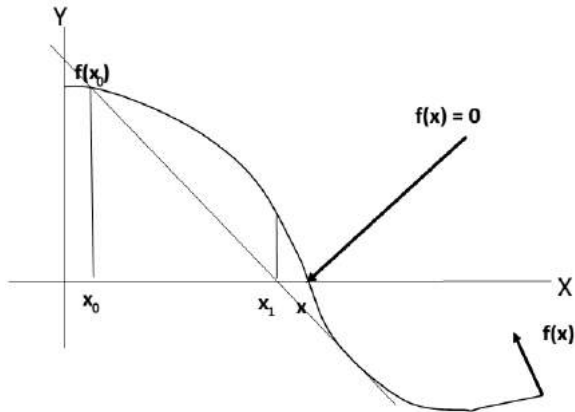
```

B. Metode Newton Raphson

Metode *Newton Raphson* adalah Metode Terbuka. Metode yang tidak memilih interval yang memuat/ mengurung akar persamaan dari persamaan non linier. Adapun Algoritma Metode *Newton Raphson*:

- a. Pilih sembarang nilai awal x_0 .
- b. Melalui titik $(x_0, f(x_0))$, tarik suatu garis singgung kurva sehingga memotong sumbu x . Titik potong garis singgung dengan sumbu x tersebut disebut x_1 .
- c. Melalui titik $(x_1, f(x_1))$, tarik kembali garis singgung kurva sehingga memotong sumbu x . Titik potong garis singgung dengan sumbu x tersebut disebut x_2 .
- d. Proses pembuat garis singgung dan penentuan titik potong garis singgung dilanjutkan hingga diperoleh $f(x_n) = 0$ atau $f(x_n)$ sedekat mungkin dengan 0 (nol).





Gambar 1.6 Model Fungsi Metode Newton Raphson

Untuk mendapatkan persamaan iteratif dari metode Newton Repshon perhatikan model geometris berikut. Dari persamaan umum garis melalui titik (a,b) dengan gradient m,

$$y - b = m(x - a) \Rightarrow x = \frac{y - b}{m} + a = \frac{y - f(x_i)}{f'(x_i)} + x_i$$

Jika $x = x_{i+1}$ dan $y = 0$. sehingga diperoleh : $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$

Untuk kasus pada metode newton rapshon di atas, maka jika $x = x_{i+1}$, maka $y = 0$ sehingga diperoleh $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$. persamaan: $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ disebut *formula*.

Konstruksi Scribs m-funciotn Metode Newton Raphson

```
function [H]=NR(f,X,e)
syms x
df=diff(f,x);
H=[{'f(x)=' , char(f)}; {'f'(x)=' , char(df)}];
H=[H; {'x_0 = ' , num2str(X)}];
if abs(eval(subs(df,X)))>=1;
    H=[H; {'e = ' , num2str(e)}];
    c=1;
    while abs(eval(subs(f,X(c))))>e;
        ho=[{' '}; {'Iterasi ke-' , num2str(c)}];
```



```

h1={(['x(',num2str(c),'') = x(',num2str(c-1),'') -
f(x',...
num2str(c-1),'')/df(x',num2str(c-1),'')]);
h2={([' = ',num2str(X(c),'') - f(',num2str(X(c),'...
')')/df(',num2str(X(c),'')')]);
X(c+1)=X(c)- subs(f,X(c))/subs(df,X(c));
F=eval(subs(f,X));
E(c)=abs(F(c+1)-F(c));
DF=eval(subs(df,X));
h3={([' = ',num2str(X(c),'') - ('',num2str(F(c),'')/('',...
num2str(DF(c),'')')]);
h4={([' =
',num2str(X(c),'')-
(',num2str(F(c)/DF(c),'')')]);
h5={([' = ',num2str(X(c+1)))];
h6={(['f(x',num2str(c),'') = ',num2str(F(c+1))])];
c=c+1;
if abs(F(c))>e
h7={(['Karena
|f(x',num2str(c-
1),'')|>',num2str(e),...
' = e maka iterasi berlanjut']]);
else
h7={(['Karena
|f(x',num2str(c-
1),'')|<',num2str(e),...
' = e maka iterasi berakhir']]);
end
H=[H;ho;h1;h2;h3;h4;h5;h6;h7];
end
subplot(2,1,1)
plot(E,'-r')
grid on
xlabel('Iterasi')
ylabel('|error|')
title('Grafik konvergensi Newton_Raphson')
subplot(2,1,2)
hold on
plot(X(c),F(c),'*r')
hold on
ezplot(f,[X(c)-3 X(c)+3])
legend(['x = ',num2str(X(c))],[f(x) = ',num2str(F(c))])
hold on
grid on
else
H=[H;{'Maaf, pilihan xo tidak konvergen'}];
end

```



Contoh Simulasi:

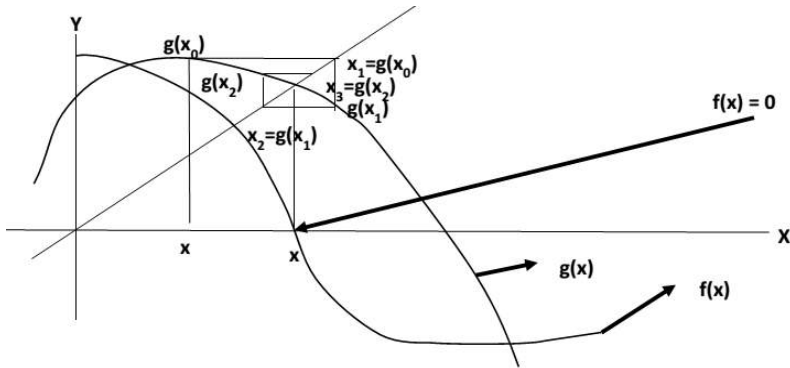
```
>> NR(f,-0.28125,0.001)
ans =
    'f(x)=2*x*exp(1 - x) + 2'
    'f'(x)=2*exp(1 - x) - 2*x*exp(1 - x) '
    'x0 = -0.28125'
    'e = 0.001'
    ' '
    'Iterasi ke-1'
    'x(1) = x(0) - f(x0)/df(x0) '
    ' = -0.28125 - f(-0.28125)/df(-0.28125) '
    ' = -0.28125 - (-0.02564)/(9.2279) '
    ' = -0.28125- (-0.0027786) '
    ' = -0.27847'
    'f(x1) = -6.3339e-05'
    'Karena |f(x1)|<0.001 = e maka iterasi berakhir'
```

C. Metode Fixet Point

Metode *Fixet Point* atau titik tetap disebut juga sebagai metode iterasi sederhana merupakan cara menentukan nilai x sehingga $f(x) = 0$ (akar fungsi) dengan cara menentukan nilai x sehingga garis $y=x$ berpotongan dengan suatu fungsi $y = g(x)$ dimana fungsi $y = g(x)$ tersebut di bentuk persamaan solusi x dari fungsi $f(x)$. Adapun Algoritma Metode *Fixet Point*:

- a. Pilih sembarang nilai awal x_0 .
- b. Kemudian tentukan nilai $g(x_0)$ dimana nilai ini sama dengan x_1 , karena $g(x_0)$ berada sejajar dengan $y = x_1$.
- c. Selanjutnya dari x_1 ditentukan nilai $g(x_1)$ dimana nilai tersebut sama dengan x_2 . hal ini disebabkan karena titik $g(x_1)$ sejajar dengan $y = x_2$.
- d. Proses $x_n = g(x_{n-1})$ ini dilakukan hingga nilai $f(x_n) = 0$ atau dekat dengan 0 (nol).





Gambar 1.7 Model Fungsi Metode *Fixet Point*

Konstruksi *Scribs* M-Function Metode *Fixet Point*

```
function [H] = secant(f,x1,x2,e)
syms x;
X=[x1 x2];
H=[{'f(x)=',char(f)}];
H=[H;{'[x1 x2] = ',num2str(x1),' ',num2str(x2),' '}]];
H=[H;{'e = ',num2str(e)}];
Fx=eval(subs(f,X));c=2;
while abs(Fx(end))>e;
    H=[H;{' '};{'Iterasi ke-',num2str(c-1)}];
    b=eval(subs(f,X(c)));
    g=eval(subs(f,X(c-1)));
    X(c+1)=X(c)-(X(c)-X(c-1))*b/(b-g);
    H=[H;{'x',num2str(c+1),' = ',num2str(X(c+1)),' -',
    (x',num2str(c),' -x',num2str(c-1)),
    num2str(c-1),' f(x',num2str(c-1)),
    num2str(c-1),' ) / f(x',num2str(c-1)),
    num2str(c-1),' )'}];
    H=[H;{' = ',num2str(X(c)), '-(',num2str(X(c)),
    num2str(X(c-1)),') (',num2str(b)),
    num2str(c-1),' ) / (',num2str(b)),
    num2str(c-1),' )'}];
    H=[H;{' = ',num2str(X(c)), '-(',num2str(X(c)-X(c-1)),
    num2str(c-1)),') (',num2str(b)),
    num2str(c-1),' ) / (',num2str(b-g)),
    num2str(c-1),' )'}];
    H=[H;{' = ',num2str(X(c+1))}]];
    Fx=eval(subs(f,X));
    if abs(Fx(end))>e;
        h=[{'|f(x',num2str(c+1))| = ',num2str(abs(Fx(end))),
        num2str(c+1),' > ',num2str(e),
        ' = e, maka Lanjut'}];
    else
        h=[{'|f(x',num2str(c+1))| = ',num2str(abs(Fx(end))),
        num2str(c+1),' < ',num2str(e),
        ' = e, maka Berhenti'}];
    end
end
```



```

',...
        num2str(abs(Fx(end))), ' < ', num2str(e), ' = e, maka
Stop']});
    end
    c=c+1;
    H=[H;h];
end
H=[H;{' '};{(['Jadi akar persamaan adalah x =
', num2str(X(end)), ...
' dengan f(x) = ', num2str(Fx(end))])}]);
subplot(2,1,1);
plot(abs(Fx), '-*r');
grid on;
xlabel('Iterasi');
ylabel('Jarak f(x) dengan 0 ');
title('Grafik konvergensi error Secant');
subplot(2,1,2);
plot(X(end), Fx(end), '*r');
hold on;
ezplot(f, [X(end)-3 X(end)+3]);
grid on;
legend([' x = ', num2str(X(end))], ['f(xr) = ', num2str(Fx(end))]);
end
''

```

Contoh Simulasi:

```

>> g=-exp(x-1)
g =
-exp(x - 1)
>> FP(f,g,-0.28125,0.001)
ans =
'f(x)=2*x*exp(1 - x) + 2'
'g'(x)=-exp(x - 1)'
'X = -0.28125'
'e =0.001'
' '
'Iterasi ke-1'
' x1 = g(x0) = -0.27769'
'f(x1) = f(-0.27769) = 0.0071074'
'Karena |f(x1)| > 0.001 = e maka proses lanjut'
' '
'Iterasi ke-2'
' x2 = g(x1) = -0.27868'
'f(x2) = f(-0.27868) = -0.0019817'
'Karena |f(x2)| > 0.001 = e maka proses lanjut'
'Iterasi ke-3'
' x3 = g(x2) = -0.2784'
'f(x3) = f(-0.2784) = 0.00055163'
'Karena |f(x3)| < 0.001 = e maka proses berhenti'

```



Uji Kompetensi 1



1. Buatlah *scribs* menggunakan struktur *Swicth Case* untuk menggabungkan semua simulasi yang sudah Anda buat baik metode terbuka maupun metode tertutup sesuai ketentuan berikut.

Cabang/Pilihan	Sub Cabang (Case)
Metode Tertutup	1. <i>Bisection</i> 2. <i>Regula Falsi</i>
Metode Terbuka	1. <i>Secant</i> 2. <i>Newton Raphson</i> 3. <i>Fixet Point</i>

Contoh konstruksi *scribs*:

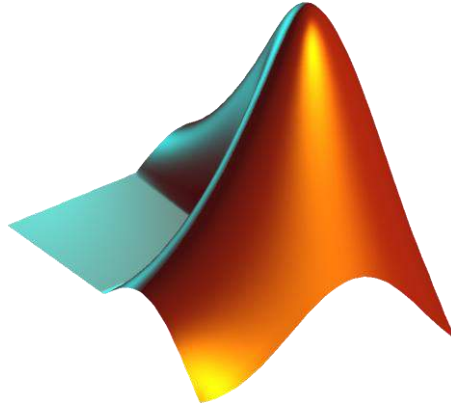
```
clc;clear all;
disp('Program   : Simulasi Switch Case');
disp('Programer  : Syaharuddin');
disp('Cabang:');
disp('Daftar Pilihan Cabang');
p=input('Pilihan = ');
disp('=====');
switch p
    case 1
        disp('Sub Cabang');
        q=input('Pilihan = ');
        switch q
            case 1
                scribs
            case 2
                scribs
        end
    case 2
        disp('Sub Cabang');
        q=input('Pilihan = ');
        switch q
            case 1
                scribs
            case 2
                scribs
        end
end
```




```
case n  
end
```

2. Buatlah *scribs* untuk menyelesaikan persoalan sistem persamaan non linier menggunakan metode *Jacobian*, *Gauss Seidel*, dan *Newton Raphson*.





BAB 2

GUI MATLAB PNL & SPNL

Tujuan Perkuliahan:

1. Mahasiswa mampu membuat *interface (guide)* program aplikasi sederhana.
2. Mahasiswa mampu mengembangkan *interface* untuk berbagai struktur bahasa pemrograman dalam menyelesaikan persoalan numerik.
3. Mahasiswa mampu membuat *MainMenu* untuk menggabungkan program aplikasi.

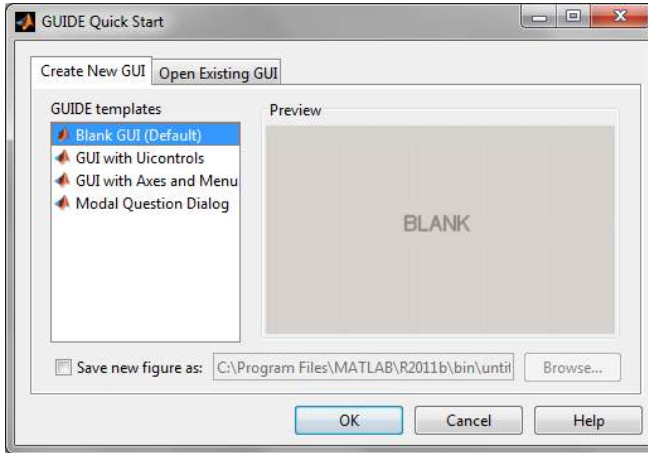
2.1 Pengantar GUI

Guide adalah salah satu komponen dari Matlab untuk membuat *interface (desain form)* proses penyelesaian persoalan matematika yang lebih efisien dan menarik. Tidak seperti m-file hanya bisa bermain di *Command Windows*. Di sini, Anda akan membuat *form* (lembar kerja) untuk masing-masing program aplikasi dengan menggunakan atribut yang sudah disediakan oleh Matlab. Silahkan Anda buka Matlab dan ketikkan di *Command Windows* scrib berikut:

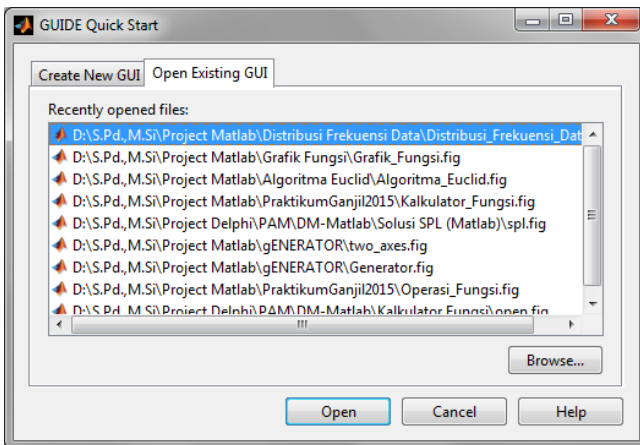


>> guide

Kemudian tekan **Enter**, maka akan muncul kotak dialog berikut ini.



(a)




(b)

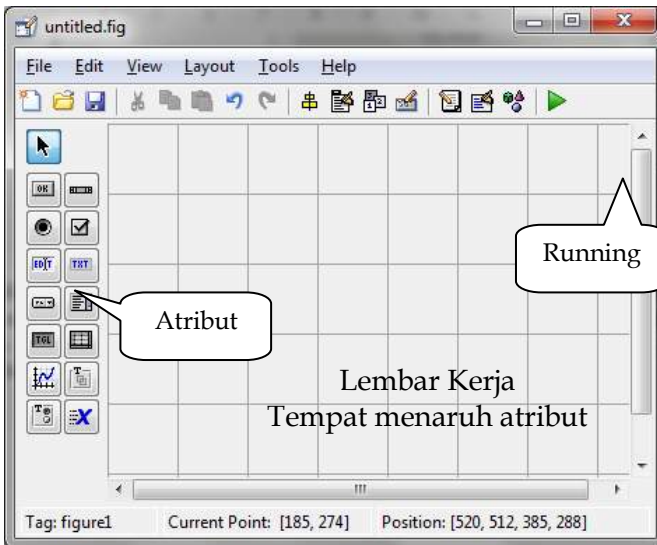
Gambar 2.1 Tampilan Awal Guide, (a) *Create New GUI*, (b) *Open Existing GUI*

Pada gambar di atas terdapat 2 menu yaitu (1) *Create New GUI* yakni untuk membuat *guide* baru dan (2) *Open Existing GUI*



yakni untuk membuka *guide* yang sudah pernah Anda buat sebelumnya yang tersimpan dalam PC/Labtop Anda. Jika Anda ingin membuat *Guide* baru, silahkan pilih **Blank GUI (Default)**, kemudian pilih **OK**. Maka akan muncul tampilan *Guide* kosong dan Anda siap membuat program aplikasi dengan menggunakan atribut yang ada. Untuk mengetahui nama-nama atribut, silahkan Anda jalankan kursor maka akan muncul dengan sendirinya *hint* dari masing-masing atribut. Anda cukup klik sekali di atribut, kemudian klik sekali lagi di lembar kerja maka atribut tersebut siap dipakai.

Sebelum lebih lanjut Anda membuat *guide*, silahkan simpan program aplikasi Anda terlebih dahulu dengan meng-klik tombol simpan  pada *guide* atau klik **File** → **Save** → **OK**. Tapi ingat menulis nama porgram aplikasi tidak boleh spasi, harus menggunakan underscore (garis bawah) jika terdiri dari 2 kata atau lebih. Cara menyimpan yang lain adalah melakukan running dengan menekan tombol **F5** di keyboard Labtop/PC Anda.











Gambar 2.2 *Guide* Kosong

Adapun fungsi dari masing-masing atribut: dapat dilihat pada tabel berikut ini.







Tabel 2.1 Atribut GUI Matlab dan Fungsinya













No	Gambar	Nama (<i>Hint</i>)	Fungsi
1		<i>Push Button</i>	Sebagai tombol (proses, hapus, keluar, dll)
2		<i>Slider</i>	Untuk minimumkan tampilan jika butuh layar lebar
3		<i>Radio Button</i>	Untuk meletakkan pilihan
4		<i>Check Box</i>	Untuk meletakkan pilihan
5		<i>Edit Text</i>	Sebagai tempat input atau output
6		<i>Static Text</i>	Sebagai label/nama properti yang digunakan
7		<i>Pop-up Menu</i>	Sama dengan <i>Check Box</i> , hanya pilihan di <i>hint</i> .
8		<i>List Box</i>	Sebagai output dalam jumlah banyak string.
9		<i>Toggle Button</i>	Sama dengan <i>Push Button</i>
10		<i>Table</i>	Membuat output dalam bentuk tabel.
11		<i>Axes</i>	Untuk menggambar grafik/histogram
12		<i>Panel</i>	Untuk menyatukan atribut dalam satu kelompok
13		<i>Button Group</i>	Untuk menyatukan atribut dalam satu kelompok
14		<i>ActiveX Control</i>	Untuk memunculkan beberapa control penting.

Adapun fungsi beberapa sub toolbar dari *Guide* Matlab sebagai berikut:

Tabel 2.2 SubToolbar GUI Matlab

No	Gambar	Nama (<i>Hint</i>)	Keyboard	Fungsi
1		<i>New Figure</i>	Ctrl+N	Membuat <i>guide</i> baru
2		<i>Open Figure</i>	Ctrl+O	Membuka <i>guide</i> yang sudah ada.
3		<i>Save Figure</i>	Ctrl+S	Menyimpan <i>guide</i>
4		<i>Cut</i>	Ctrl+X	Memindahkan atribut dalam <i>guide</i>



5		<i>Copy</i>	Ctrl+C	Mengkopi atribut dalam <i>guide</i>
6		<i>Paste</i>	Ctrl+V	Menempelkan atribut dalam <i>guide</i>
7		<i>Redo</i>	Ctrl+Z	Mengulang mundur perintah
8		<i>Undo</i>	Ctrl+Y	Mengulang maju perintah
9		<i>Align Object</i>	-	Mengatur tata letak atribut
10		<i>Menu Editor</i>	-	Membuat main menu (toolbar)
11		<i>Tab Order Editor</i>	-	Mengunci/mencari atribut
12		<i>Tool Bar Editor</i>	-	Menambahkan toolbar ke <i>guide</i>
13		<i>Editor</i>	-	Menampilkan editor
14		<i>Property Inspector</i>	-	Mengganti/mengatur karakter atribut
15		<i>Object Browser</i>	-	Mencari atribut
16		<i>Run Figure</i>	Ctrl+T / F5	Melakukan eksekusi / running

Dalam pembuatan program aplikasi khususnya di bagian III ini, Anda akan banyak menggunakan atribut: *Push Button*, *Static Text*, *Edit Text*, *ListBox*, *Axes*, *Table*, dan *GroupBox*. Sedangkan untuk *Tollbar* yang sering akan digunakan adalah *New*, *Save*, *Menu Editor*, *Tool Bar Editor*, dan *Run*.

2.2 GUI Sederhana

Struktur urut adalah struktur paling sederhana dalam bahasa pemrograman. Karena step dari struktur terdiri dari:

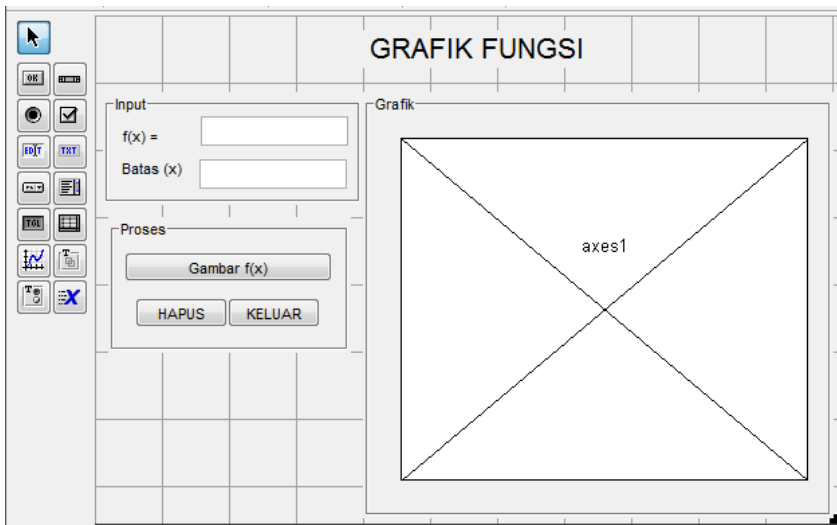
1. Definisikan Fungsi/Variabel Inputan
2. Rumus (perintah)
3. Definisikan *Output*

Hal ini tidak seperti struktur yang lain harus melakukan kontrol terlebih dahulu.

Program Aplikasi	:	Grafik Fungsi
Atribut	:	StaticText (3), Edit Text (2), Push Button (3), Button Group (3), Axes (1)
Manfaat	:	Menggambar grafik pada interfal tertentu.

Langkah-langkah Kerja:

1. Desainlah *guide* seperti berikut:



2. Pilih tombol “Gambar F(x)” kemudian klik kanan, pilih *View Callbacks* → *Callbacks*, maka akan muncul m-file. Ketikkan *scribs* berikut ini.

```
function pushbutton2_Callback(hObject, eventdata, handles)
syms x
f=get(handles.edit1,'String');
d=str2num(get(handles.edit2,'String'));
hold off
ezplot(f,d)
grid on
```

3. Pilih tombol “Hapus” kemudian klik kanan, pilih *View Callbacks* → *Callbacks*. Ketikkan *scribs* berikut ini.



```
function pushbutton3_Callback(hObject, eventdata, handles)
set(handles.edit1,'String','');
set(handles.edit2,'String','');
hold off
plot(0,0)
```

- Pilih tombol “Keluar” kemudian klik kanan, pilih *View Callbacks* → *Callbacks*. Ketikkan scribs berikut ini.

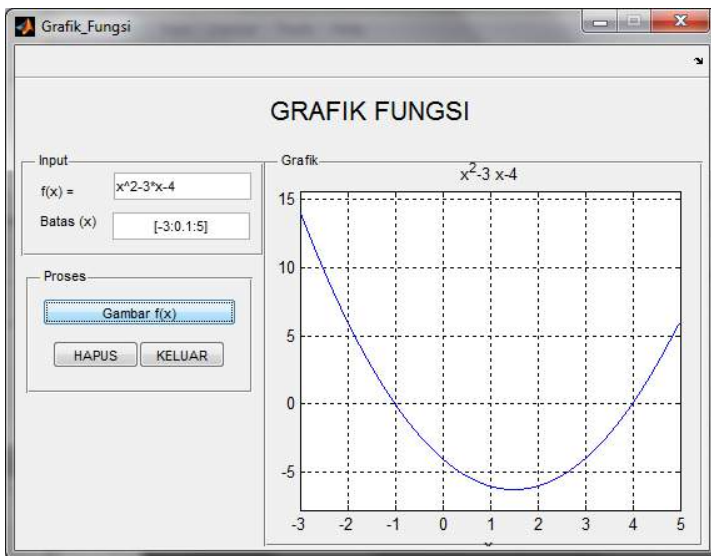
```
function pushbutton4_Callback(hObject, eventdata, handles)
close;
```

- Silahkan tekan F5 atau klik tombol *running*. Kemudian simulasikan fungsi berikut ini.

$$F(x) = x^2 - 3x - 4$$

Batas = [-3:0.1:5] ; dibaca: Batas Cartesius dari -3 sampai 5 dengan delta x = 0.1

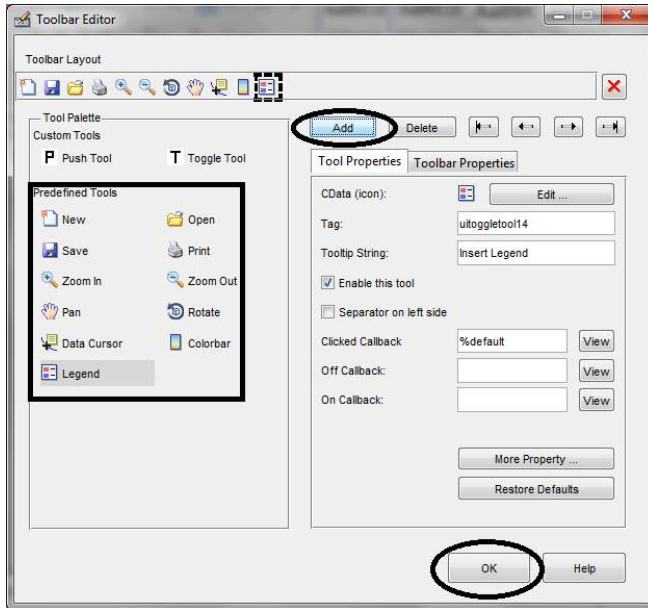
Maka akan tampil seperti berikut ini. (Perhatikan cara input fungsi dan interval).



Gambar 2.3 Simulasi Grafik Fungsi



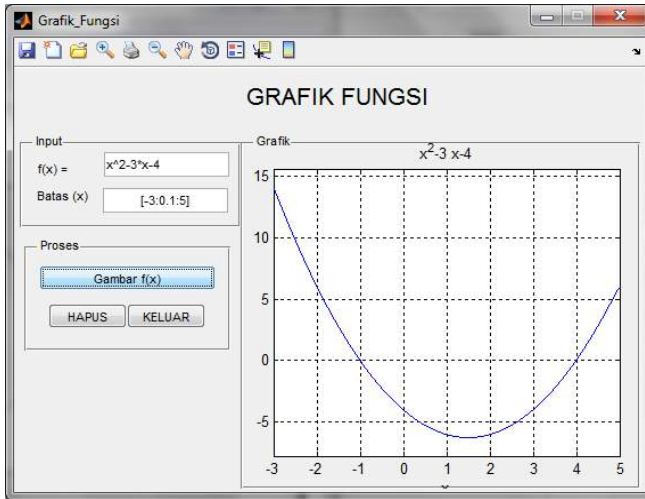
6. Tampilkanlah *Toolbar* pada *guide* Anda agar bisa melakukan banyak modifikasi. Silahkan klik *Toolbar Editor*, maka akan muncul kotak dialog berikut.



Gambar 2.4 Penambahan TollBar

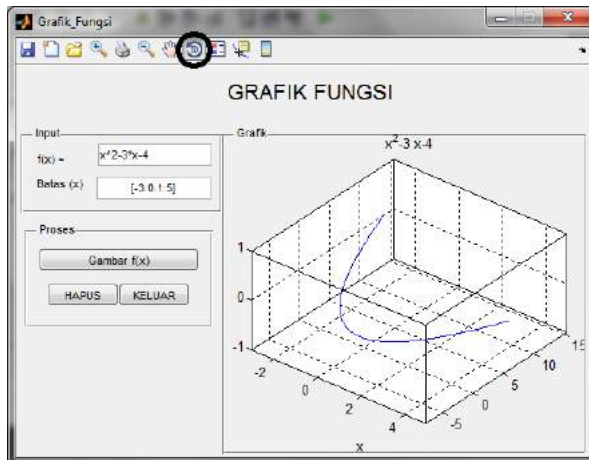
7. Pilih item di *Predefined Tools*, kemudian klik *Add*, terakhir klik *OK*. Silahkan *running* program aplikasi Anda, kemudian lakukan simulasi dengan fungsi tadi.





Gambar 2.5 Tampilan Baru Setelah Penambahan TollBar

8. Kemudian silahkan pilih *Rotate 3D*, klik pada grafik maka akan diperoleh output seperti berikut.



Gambar 2.6 Fungsi TollBar Rotasi untuk Memutar Grafik

9. Silahkan lakukan simulasi dengan berbagai macam fungsi yang lain. Selamat mencoba.



A. Metode Bisection

1. Mengambil Atribut

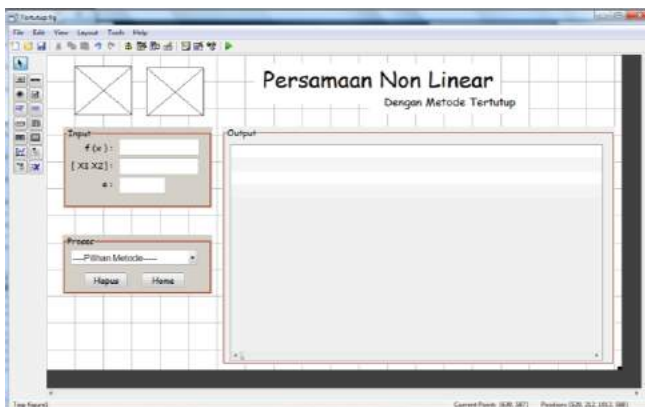
Periksa atribut GUI Matlab dan letakkan di lembar kerja GUI.

Atribut	Jumlah
<i>Label</i>	5
<i>Edit</i>	3
<i>Button</i>	2
<i>Panel</i>	3
<i>Pop-up Menu</i>	1
<i>Axes</i>	2

Untuk bagian gambar (*Axes*) penulis menyiapkan logo lembaga. Jika anda ingin mengganti, Anda dapat menggunakan logo sesuai keinginan Anda.

2. Mengatur Desain GUI

Susunlah atribut di atas sesuai dengan gambar di bawah ini!



Gambar 2.7 GUI Awal PNL Metode Tertutup



3. Konstruksi M-Function di Editor

Berikunya, Anda membuat file m-function seperti berikut, pertama *m-function* Metode *Bisection*, kedua *m-function* Metode *Regula Falsi*.

M-Function Metode Bisection

```
function [H] = bisection(f,x1,x2,e)
syms x;
H=[{'f(x)=',char(f)}]
H=[H;{'[x1 x2] = ',num2str(x1),' ',num2str(x2),' '}]
H=[H;{'e =',num2str(e)}]
f1=eval(subs(f,x1))
f2=eval(subs(f,x2))
H=[H;{'f(',num2str(x1),' ) = ',num2str(f1)}]
H=[H;{'f(',num2str(x2),' ) = ',num2str(f2)}]
if f1*f2<0;
    H=[H;{'Karena f(x1)f(x2)<0, maka x shg f(x)= 0
pada interval [' ,...
        num2str(x1),' ',num2str(x2),' ']}];
    fr=1;c=0;
    while abs(fr)>e;
        c=c+1;
        H=[H;{' '};{'Iterasi ke-',num2str(c)}];
        xr=0.5*(x1+x2);
        fr=eval(subs(f,xr));
        H=[H;{' xr = 0.5(x1 + x2)'}];
        H=[H;{' = 0.5(',num2str(x1),' +
',num2str(x2),' ')}];
        H=[H;{' = 0.5(',num2str(x1+x2),' ')}];
        H=[H;{' = ',num2str(xr)}];
        H=[H;{'f(xr)= ',num2str(fr)}];
        E(c)=abs(fr);
        if abs(fr)>e;
            H=[H;{'Karena |f(xr)| =
',num2str(abs(fr)), '>',num2str(e), ...
            '=e maka update [x1 x2]'}];
            f1=eval(subs(f,x1));
            if fr*f1<0;
                x2=xr;
                h1=[{'Karena f(x1) = ',num2str(f1),'
maka f(x1)f(xr)<0, shg x2=xr'}];
                h2=[{'Jadi interval baru adalah
[' ,num2str(x1), ...
            ' ',num2str(x2),' ']}];
            else
                x1=xr;
                h1=[{'Karena f(x1) = ',num2str(f1),'
```



```

maka f(x1)f(xr)>0, shg x1=xr']]);
        h2=[(['Jadi interval baru adalah
[' ,num2str(x1), ...
        ' ',num2str(x2), '']]);
        end
        H=[H;h1;h2];
    else
        H=[H;{(['Karena |f(xr)| =
',num2str(abs(fr)), '<',num2str(e), ...
        '=e maka proses berhenti']]);
        end
    end
    H=[H;{ ' ' };{(['Jadi akar persamaan adalah x =
',num2str(xr), ...
    ' dengan f(xr) = ',num2str(fr)])}];
    subplot(2,1,1)
    plot(E, '-*r')
    grid on
    xlabel('Iterasi')
    ylabel('Jarak f(x) dengan 0 ')
    title('Grafik konvergensi error Bisection')
    subplot(2,1,2)
    plot(xr, fr, '*r')
    hold on
    ezplot(f, [xr-3 xr+3])
    grid on
    legend([' x = ', num2str(xr)], ['f(xr)=
', num2str(fr)])
else
    H=(['Karena f(x1)f(x2)>0, maka tidak ada x shg
f(x)=0 pada interval [' , ...
    num2str(x1), ' ', num2str(x2), '']]);
end
end
end

```

M-Function Metode Regula Falsi

```

function [H] =RF(f,x1,x2,e)
syms x;
H=[(['f(x)=', char(f)])];
H=[H;{(['x1 x2] = [' ,num2str(x1), ' ', num2str(x2), '']]);
H=[H;{(['e =', num2str(e)])];
f1=eval(subs(f,x1));
f2=eval(subs(f,x2));
H=[H;{(['f(', num2str(x1), ') = ', num2str(f1)])};
H=[H;{(['f(', num2str(x2), ') = ', num2str(f2)])};
if f1*f2<0
    H=[H;{(['Karena f(x1)f(x2)<0, maka x shg f(x)= 0
pada interval [' , ...
        num2str(x1), ' ', num2str(x2), '']]);

```



```

fr=1;c=0;
while abs(fr)>e;
    c=c+1;
    H=[H;{' '};{(['Iterasi ke-',num2str(c)])}];
    f1=eval(subs(f,x1));f2=eval(subs(f,x2));
    xr=x1-(x2*f1-x1*f1)/(f2-f1);
    fr=eval(subs(f,xr));
    H=[H;{(['xr = x1 - (x2-x1)f(x1)/(fx(2)-
f(x1))'])}];
    H=[H;{([' = ',num2str(x1),'-(',num2str(x2),' -
',num2str(x1),...
    ')f(',num2str(x1),')/(f(',num2str(x2),'-
f(',num2str(x1),'))'])}];
    H=[H;{([' = ',num2str(x1),'-(',num2str(x2-
x1),...
    ')(',num2str(f1),')/(',num2str(f2),' -
',num2str(f1),'))'])}];
    H=[H;{([' = ',num2str(x1),'-(',num2str((x2-
x1)*f1),')/(',num2str(f2-f1),'))'])}];
    H=[H;{([' = ',num2str(x1),'-(',num2str((x2-
x1)*f1/(f2-f1),'))'])}];
    H=[H;{([' = ',num2str(xr)])}];
    H=[H;{(['f(xr)= ',num2str(fr)])}];
    E(c)=abs(fr);
    if abs(fr)>e;
        H=[H;{(['Karena |f(xr)| =
',num2str(abs(fr)), '>',num2str(e),...
        '=e maka update [x1 x2]'])}];
        f1=eval(subs(f,x1));
        if fr*f1<0;
            x2=xr;
            h1=[(['Karena f(x1) = ',num2str(f1),' maka
f(x1)f(xr)<0, shg x2=xr'])];
            h2=[(['Jadi interval baru adalah
',num2str(x1),...
            ' ',num2str(x2),')'])];
        else
            x1=xr;
            h1=[(['Karena f(x1) = ',num2str(f1),' maka
f(x1)f(xr)>0, shg x1=xr'])];
            h2=[(['Jadi interval baru adalah
',num2str(x1),...
            ' ',num2str(x2),')'])];
        end
        H=[H;h1;h2];
    else
        H=[H;{(['Karena |f(xr)| =
',num2str(abs(fr)), '<',num2str(e),...
        '=e maka proses berhenti'])}];
    end
end

```



```

end
H=[H;{(['Jadi akar persamaan adalah x =
',num2str(xr),...
      ' dengan f(xr) = ',num2str(fr)])}];
subplot(2,1,1)
plot(E,'-r')
grid on
xlabel('Iterasi')
ylabel('Jarak f(x) dengan 0 ')
title('Grafik konvergensi error Regula_Falsi')
subplot(2,1,2)
plot(xr,fr,'*r')
hold on
ezplot(f,[xr-3 xr+3])
grid on
legend([' x = ',num2str(xr)],['f(xr)=
',num2str(fr)])
else
H=[(['Karena f(x1)f(x2)>0, maka tidak ada x shg
f(x)=0 pada interval [' ,...
      num2str(x1), ' ',num2str(x2), ''])]];
end
end
end

```

Selanjutnya, perhatikan GUI Anda dan ikuti langkah berikut ini.

- a. Klik kanan pada atribut *PopupMenu*, pilih *View Callbacks* → *Callbacks*.
- b. Silahkan ketikkan *scribs* berikut.

```

function popupmenu1_Callback(hObject, eventdata,
handles)
syms x
f=get(handles.edit1,'string');
%g=get(handles.edit2,'string');
A=str2num(get(handles.edit2,'string'));
x1=A(1);
x2=A(2);
%X=str2double(get(handles.edit4,'string'));
e=str2double(get(handles.edit3,'string'));
p=get(handles.popupmenu1,'Value');
switch p
case 1
case 2
figure
H=bisection(f,x1,x2,e);
set(handles.uitable1,'data',H);
case 3

```



```

figure
H=RF(f,x1,x2,e);
set(handles.uitable1,'data',H);
end

```

c. Pada tombol **Hapus**, ketikkan *scrips* berikut.

```

set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.uitable1,'data',[]);
%hold off
%subplot(2,1,1)
%plot(0,0)
%hold off
%subplot(2,1,2)
%plot(0,0)

```

d. Pada tombol **Home**, ketikkan *scrips* berikut.

```

function pushbutton1_Callback(hObject, eventdata,
handles)
h=waitbar(0,'Silahkan Menunggu.....');
for i=1:300
    waitbar(i/300);
end
close(h);
close(Tertutup);
Interface;

```

Tombol *Home* ini nantinya akan berfungsi pada saat Anda akan membuat Tampilan Utama dalam menggabungkan semua metode dalam satu GUI.

e. Tekan tombol RUN berwarna hijau paling atas GUI Anda atau bisa juga tekan tombol F5 pada keyboard.

4. Simulasi

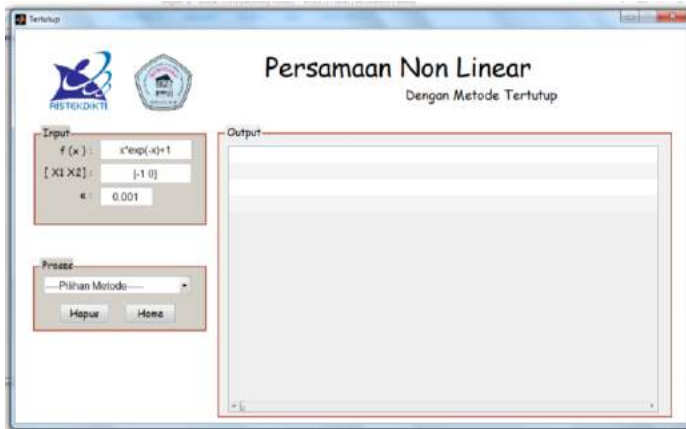
Misalkan Anda akan menentukan solusi dari pertidaksamaan linier berikut:

$f(x) = e^{-x} + 1$, dengan error (e) = 0,001 pada interval nilai awal [-1, 0].

Jalankan program, isikan fungsi non linier di atas pada label “f(x)”, titik awal yang digunakan pada label “[x1,x2]”, dan

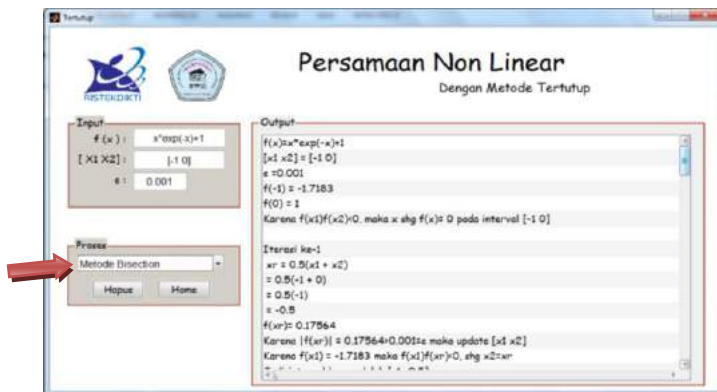


nilai error pada label “e”. tampilan seperti gambar di bawah ini.



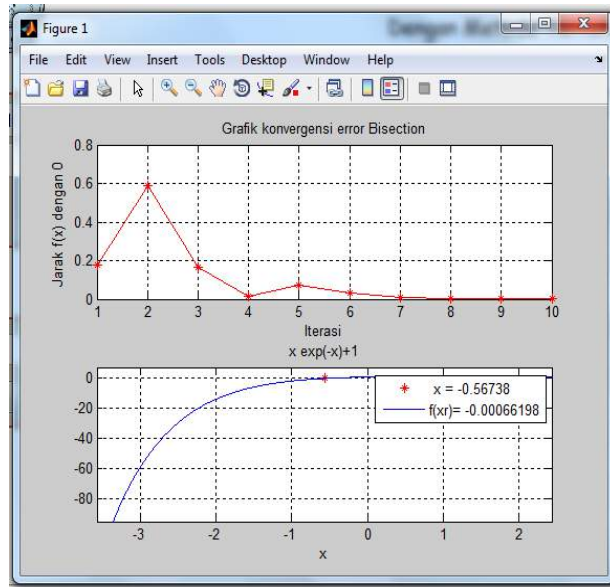
Gambar 2.8 Hasil Running GUI PNL

Pada *Panel* Proses pilih menu metode yang akan digunakan untuk menyelesaikan permasalahan Matematika yang diberikan, dalam hal ini menggunakan metode Bisection. Hasil perhitungan langsung ditampilkan pada panel *Output* di sebelah kanan dan tampilan grafik penyelesaian. Adapun tampilan seperti gambar di bawah ini.



Gambar 2.9 Simulasi Metode Bisection





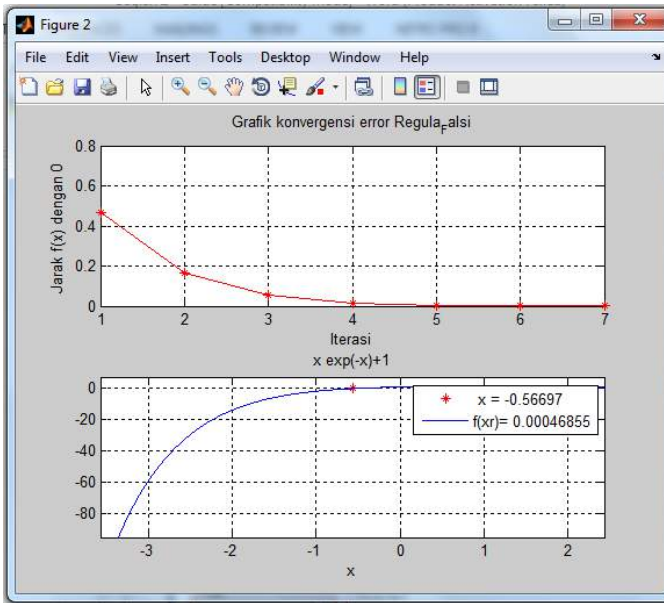
Gambar 2.10 Tampilan Grafik Metode Bisection

Terlihat bahwa hasil simulasi ditemukan nilai $f(x) = -0,00066198$ saat $x = -0,56738$ yang merupakan hasil numerik dari fungsi tersebut.

Selanjutnya, Anda bisa memilih metode kedua yakni *Regula Falsi*. Adapun tampilan hasil seperti gambar di bawah ini.



Gambar 2.11 Simulasi Metode Regula Falsi



Gambar 2.12 Grafik Metode Regula Falsi

Selanjutnya, Anda bisa mencoba dengan contoh soal lain.

2.4 PNL Metode Terbuka

Selanjutnya, kita akan mengkonstruksi GUI untuk metode terbuka dalam menyelesaikan persamaan non linier. Adapun caranya hampir sama dengan metode tertutup di atas.

A. Metode Secant

1. Mengambil Atribut

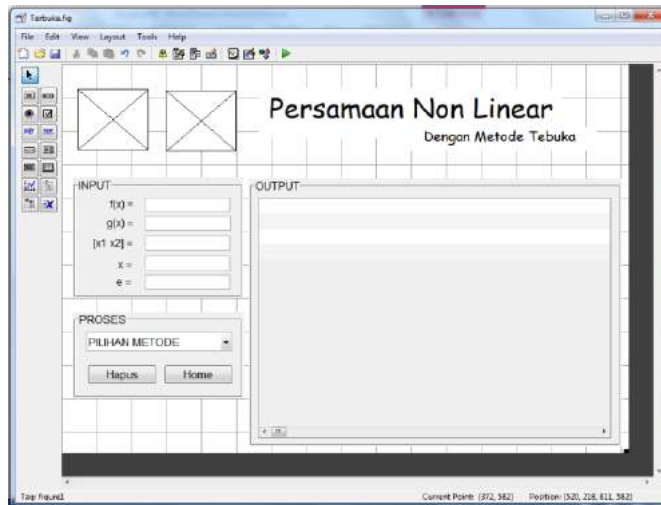
Atribut	Jumlah
Label	7
Edit	6
Button	2
Panel	3
Pop-up Menu	1
Axes	2



Untuk bagian gambar (*Axes*) penulis menyiapkan logo lembaga. Jika anda ingin mengganti, anda dapat menggunakan logo sesuai keinginan anda.

2. Mengatur Desain GUI

Susunlah atribut di atas sesuai dengan Gambar 2.13 di bawah ini!



Gambar 2.13 Tampilan GUI PNL Metode Terbuka

3. Konstruksi M-Function

Berikunya, Anda membuat file m-function seperti berikut, pertama *m-function* Metode *Secant*, kedua *Newton Raphson*, dan ketiga *Fixet Point*.

M-Function Metode Secant

```
function [H] = secant(f,x1,x2,e)
syms x;
X=[x1 x2];
H=[{'f(x)=' , char(f)}];
H=[H;{'[x1 x2] = [' , num2str(x1) , ' ' , num2str(x2) , ']' }];
H=[H;{'e = ' , num2str(e)}];
Fx=eval(subs(f,X));c=2;
while abs(Fx(end))>e;
    H=[H;{' ' }];{(['Iterasi ke-' , num2str(c-1))};
```



```

b=eval(subs(f,X(c)));
g=eval(subs(f,X(c-1)));
X(c+1)=X(c)-(X(c)-X(c-1))*b/(b-g);
H=[H;{(['x',num2str(c+1),' = x',num2str(c),'-
(x',num2str(c),'-x',...
num2str(c-
1),' )f(x',num2str(c),' )/f(x',num2str(c),'-
f(x',num2str(c-1),' )')]}];
H=[H;{([' = ',num2str(X(c)),'-(' ,num2str(X(c)),' -
',num2str(X(c-1)),' ) (' ,num2str(b),' )/(' ,num2str(b),' -
',num2str(g),' )')]}];
H=[H;{([' = ',num2str(X(c)),'-(' ,num2str(X(c)-X(c-
1)),' ) (' ,num2str(b),' )/(' ,num2str(b-g),' )')]}];
H=[H;{([' = ',num2str(X(c)),'-(' ,num2str((X(c)-X(c-
1))*b)/(b-g),' )')]}];
H=[H;{([' = ',num2str(X(c+1))]}];
Fx=eval(subs(f,X));
if abs(Fx(end))>e;
h=[{(['|f(x',num2str(c+1),' )| =
|',num2str(Fx(end)),' | = ',...
num2str(abs(Fx(end))),' > ',num2str(e),' =
e, maka Lanjut')]}];
else
h=[{(['|f(x',num2str(c+1),' )| =
|',num2str(Fx(end)),' | = ',...
num2str(abs(Fx(end))),' < ',num2str(e),' =
e, maka Stop')]}];
end
c=c+1;
H=[H;h];
end
H=[H;{' '};{(['Jadi akar persamaan adalah x =
',num2str(X(end)),' ...
' dengan f(x) = ',num2str(Fx(end))]}];
subplot(2,1,1);
plot(abs(Fx),'-*r');
grid on;
xlabel('Iterasi');
ylabel('Jarak f(x) dengan 0 ');
title('Grafik konvergensi error Secant');
subplot(2,1,2);
plot(X(end),Fx(end),'*r');
hold on;
ezplot(f,[X(end)-3 X(end)+3]);
grid on;
legend([' x = ',num2str(X(end))],['f(xr)=
',num2str(Fx(end))]);
end

```



M-Function Metode Newton Raphson

```

function [H]=NR(f,X,e)
syms x
df=diff(f,x);
H=[{'f(x)=',char(f)};{'f'(x)=',char(df)}];
H=[H;{'xo = ',num2str(X)}];
if abs(eval(subs(df,X)))>=1;
    H=[H;{'e = ',num2str(e)}];
    c=1;
    while abs(subs(f,X(c)))>e;
        ho={' '};{'Iterasi ke-',num2str(c)};
        h1={'x(',num2str(c),' ) = x(',num2str(c-1),' ) -
f(x',...
num2str(c-1),' )/df(x',num2str(c-1),' )']];
        h2={'(' ,num2str(X(c)), ' -
f(',num2str(X(c)),...
')/df(',num2str(X(c)), ' )']];
        X(c+1)=X(c)- subs(f,X(c))/subs(df,X(c));
        F=eval(subs(f,X));
        E(c)=abs(F(c+1)-F(c));
        DF=eval(subs(df,X));
        h3={'(' ,num2str(X(c)), ' -
(',num2str(F(c)), ' )/(',...
num2str(DF(c)), ' )']];
        h4={'(' ,num2str(X(c)), '-
(',num2str(F(c)/DF(c)), ' )']];
        h5={'(' ,num2str(X(c+1)))];
        h6={'f(x',num2str(c),' ) = ',num2str(F(c+1))]}];
    };
    c=c+1;
    if abs(F(c))>e
        h7={'|Karena |f(x',num2str(c-
1),' )|>',num2str(e),...
' = e maka iterasi berlanjut'}];
    else
        h7={'|Karena |f(x',num2str(c-
1),' )|<',num2str(e),...
' = e maka iterasi berakhir'}];
    end
    H=[H;ho;h1;h2;h3;h4;h5;h6;h7];
end
subplot(2,1,1)
plot(E,'-r')
grid on
xlabel('Iterasi')
ylabel('|error|')
title('Grafik konvergensi Newton_Raphson')
subplot(2,1,2)
hold on
plot(X(c),F(c),'*r')

```



```

        hold on
        ezplot(f, [X(c)-3 X(c)+3])
        legend(['x = ', num2str(X(c))], ['f(x) = ', num2str(F(c))])
        hold on
        grid on
    else
        H=[H; {'Maaf, pilihan xo tidak konvergen'}];
    end
end

```

M-Function Metode Fiket Point

```

function [H] = FP(f,g,X,e)
syms x;
F=subs(f,X);
H=[ {'f(x)=', char(f)}; {'g'(x)=', char(g)}];
H=[H; {'X = ', num2str(X)}];
H=[H; {'e = ', num2str(e)}];
c=1;
while abs(F(end))>e
    H=[H; {' '}; {'Iterasi ke-', num2str(c)}];
    H=[H; {'x', num2str(c), ' = g(x', num2str(c-1), ') = ', num2str(eval(subs(g,X(c))))}];
    X(c+1)=eval(subs(g,X(c)));
    F=eval(subs(f,X));
    H=[H; {'f(x', num2str(c), ') = f(', num2str(X(end)), ') = ', num2str(F(end))}];
    if abs(F(end))>e
        h=[ {'Karena |f(x', num2str(c), ')| > ', num2str(e), ' = e maka proses lanjut'}];
    else
        h=[ {'Karena |f(x', num2str(c), ')| < ', num2str(e), ' = e maka proses berhenti'}];
    end
    c=c+1;
    H=[H;h];
end
subplot(2,1,1)
plot((abs(F)), '-*r')
grid on
xlabel('Iterasi')
ylabel('|error|')
title('Grafik konvergensi Fixed_Point')
subplot(2,1,2)
plot(X(c),F(c), '*r')
hold on
ezplot(f, [X(c)-3 X(c)+3])
hold on
legend(['x = ', num2str(X(end))], ['f(x) = ', num2str(F(end))])

```



```
hold on
grid on
```

Selanjutnya, perhatikan GUI Anda dan ikuti langkah berikut ini.

- a. Klik kanan pada atribut *PopupMenu*, pilih *View Callbacks* → *Callbacks*.
- b. Silahkan ketikkan *scribs* berikut.

```
function popupmenu1_Callback(hObject, eventdata, handles)
syms x
f=get(handles.edit1,'string');
g=get(handles.edit2,'string');
A=str2num(get(handles.edit3,'string'));
x1=A(1);
x2=A(2);
X=str2double(get(handles.edit4,'string'));
e=str2double(get(handles.edit5,'string'));
p=get(handles.popupmenu1,'Value');
switch p
    case 1
    case 2
        figure
        H=secant(f,x1,x2,e);
        set(handles.uitable1,'data',H);
    case 3
        figure
        H=NR(f,X,e);
        set(handles.uitable1,'data',H);
    case 4
        figure
        H=FP(f,g,X,e);
        set(handles.uitable1,'data',H);
end
```

- c. Pada tombol **Hapus**, ketikkan *scribs* berikut.

```
function pushbutton1_Callback(hObject, eventdata, handles)
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.edit5,'string','');
set(handles.uitable1,'data',[]);
```



- d. Pada tombol **Home**, ketikkan *scrips* berikut.

```
function pushbutton2_Callback(hObject, eventdata, handles)
h=waitbar(0, 'Silahkan Menunggu.....');
for i=1:300
    waitbar(i/300);
end
close(h);
close(Terbuka);
Interface;
```

Tombol Home ini nantinya akan berfungsi pada saat Anda akan membuat Tampilan Utama dalam menggabungkan semua metode dalam satu GUI.

- e. Tekan tombol **RUN** berwarna hijau paling atas GUI Anda atau bisa juga tekan tombol F5 pada keyboard.

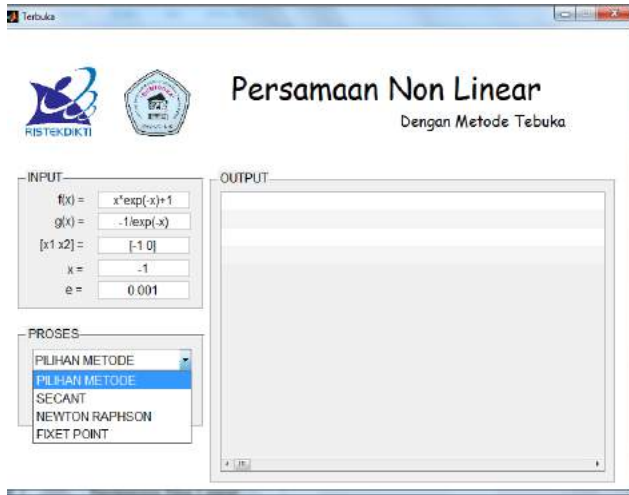
4. Simulasi

Misalkan Anda akan menentukan solusi dari pertidaksamaan linier berikut:

$f(x) = e^{-x} + 1$, dengan *error* (e) = 0,001 pada interval nilai awal $[-1, 0]$. Sedangkan untuk fungsi $g(x)$ masing-masing metode disesuaikan dengan teori yang sudah dijelaskan pada Bab 1.

Jalankan program, isikan fungsi non linier pada label “ $f(x)$ dan $g(x)$ ”, interval awal yang digunakan pada label “[x_1, x_2]” masukan titik awal pada label “ x ”, dan nilai *error* pada label “ e ”. tampilan seperti gambar di bawah ini.





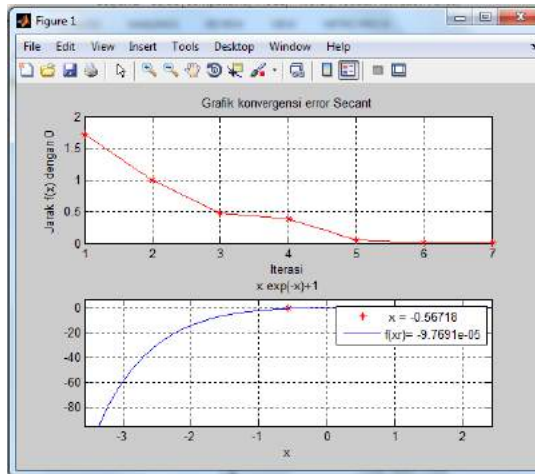
Gambar 2.14 Hasil Running GUI PNL Metode Terbuka

Pada *Panel* Proses pilih menu metode yang akan digunakan untuk menyelesaikan permasalahan yang diberikan, misalnya pertama Anda memilih metode *Secant*. Hasil perhitungan langsung ditampilkan pada panel Output di sebelah kanan dan untuk tampilan grafik penyelesaian, akan muncul jendela *figure*. Adapun tampilan seperti gambar di bawah ini.



Gambar 2.15 Simulasi Metode Secant





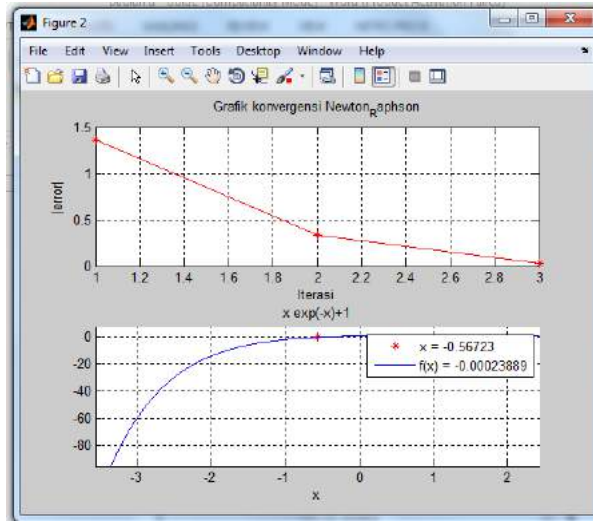
Gambar 2.16 Grafik Metode Secant

Selanjutnya pilih metode *Newton Raphson*. *Output* sesuai gambar berikut ini.



Gambar 2.17 Simulasi Metode Newton Raphson





Gambar 2.18 Grafik Metode Newton Raphson

Terakhir pilih metode *Fixet Point*. Hasil simulasi ditampilkan seperti gambar di bawah ini.



Gambar 2.19 Simulasi Metode Fixet Point

2.5 Sistem Persamaan Non Linier

Metode yang digunakan dalam menentukan solusi sistem persamaan non linier ada 2 yakni metode *Jacobian* dan Metode *Newton Raphson*.

A. Metode Jacobian

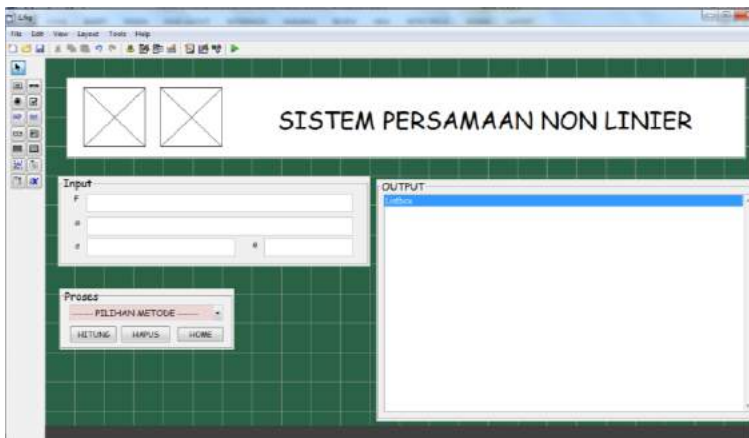
1. Mengambil Atribut

Atribut	Jumlah
<i>Label</i>	5
<i>Edit</i>	4
<i>Button</i>	3
<i>Panel</i>	3
<i>Pop-up Menu</i>	1
<i>Axes</i>	2

Untuk bagian gambar (*Axes*) penulis menyiapkan logo lembaga. Jika anda ingin mengganti, anda dapat menggunakan logo sesuai keinginan anda.

2. Mengatur Desain GUI

Susunlah atribut di atas sesuai dengan gambar di bawah ini!



Gambar 2.20 Tampilan Awal GUI SPNL



3. Konstruksi M-Function

Setelah Anda mengatur atribut sesuai gambar di atas, maka buatlah m-function sesuai langkah berikut ini.

- Klik kanan pada atribut *PopupMenu*, pilih *View Callbacks* → *Callbacks*.
- Silahkan ketikkan *scribs* berikut.

```
function popupmenu1_Callback(hObject, eventdata, handles)
    p=get(handles.popupmenu1, 'Value');
    switch p
        case 1
        case 2
            set(handles.edit2, 'Visible', 'On');
            set(handles.text3, 'Visible', 'On');
        case 3
            set(handles.edit2, 'Visible', 'On');
            set(handles.text3, 'Visible', 'On');
        case 4
            set(handles.edit2, 'Visible', 'Off');
            set(handles.text3, 'Visible', 'Off');
    end
```

- Pada tombol **Hitung**, ketikkan *scribs* berikut.

```
function pushbutton1_Callback(hObject, eventdata, handles)
    syms x y z
    F=get(handles.edit1, 'string')
    G=get(handles.edit2, 'string')
    s=str2num(get(handles.edit3, 'string'))
    e=str2double(get(handles.edit4, 'string'))
    p=get(handles.popupmenu1, 'Value')
    switch p
        case 1
        case 2
            H=({'G`='})
            J=jacobian(eval(G))
            H=[H;({'char(J)})]
            H=[H;({'G`(', num2str(s), ')='})]
            h=subst(J, s)
            %H=[H;h]
            H=[H;({'char(h)})]
            dH=eval(det(h))
            H=[H;({'|G`(', num2str(s), ')='
            ', num2str(dH)})]
            if dH>=-1&dH<=1
                m=1; c=0;
```



```

        while m>e
            h1={(['Karena |F`| = ', num2str(m), '
> ', num2str(e), ...
                ' = e, maka iterasi lanjut'])};
            h2={([''])};
            h3={(['Iterasi ke-', num2str(c+1)])};
            X=eval(subt(G,s));
            h4={(['G =
G(', num2str(s), ')=[' , num2str(X'), ''])});
            FX=eval(subt(F,X));
            m=max(abs(FX));
            if m>e
                h1={(['Karena |F`| =
', num2str(m), ' > ', num2str(e), ...
                    ' = e, maka iterasi
lanjut'])});
                h2={([''])};
                s=X;
                c=c+1;
                h3={(['Iterasi ke-
', num2str(c+1)])});
                X=eval(subt(G,s));
                h4={(['G = G(', num2str(s), ') =
[', num2str(X'), ''])});
                FX=eval(subt(F,X));
                h5={(['F = F(', num2str(X'), ') =
[', num2str(FX'), ''])});
                m=max(abs(FX));
            end
            H=[H;h1;h2;h3;h4;h5];
        end
        H=[H;{(['Iterasi berhenti'])});
    else
        H=[H;{(['Nilai awal anda salah'])});
    end
    set(handles.listbox1,'string',H)

case 3
    H={(['G` = '])};
    J=jacobian(eval(G));
    H=[H;{([char(J)])});
    h=subt(J,s);
    H=[H;{([char(h)])});
    dH=eval(det(h));
    m=1;c=0;
    if dH>=-1&dH<=1
        m=1;c=0;
        while m>e
            h1={(['Karena |F`| = ', num2str(m), '
> ', num2str(e), ...

```



```

        ' = e, maka iterasi lanjut']));
    h2={([' ']);
    h3={(['Iterasi ke-',num2str(c+1)]});
    h4={([' ']);
    for i=1:length(s)
        S=eval(subs(G,s(1)));
        for j=2:length(s)
            S=eval(subs(S,s(j)));
        end
        s(i)=S(i);
    end
    FX=eval(subt(F,s));
    h5={(['F =
F(',num2str(s),')=[',num2str(FX),' ']);
    m=max(abs(FX));
    c=c+1;
    H=[H;h1;h2;h3;h4;h5];
end
H=[H;{([' '])});
H=[H;{(['Karena |F`| = ',num2str(m),' >
',num2str(e),...
        ' = e, maka iterasi berhenti'])});
else
    H=[H;{(['Maaf nilai awal anda
keliru'])});
end
set(handles.listbox1,'string',H)
case 4
    H={([char(F)])};
    H=[H;{(['F` = '])});
    J=jacobian(eval(F));
    H=[H;{([char(F)])});
    H=[H;{(['F`(',num2str(s),') = '])});
    m=1;c=0;
    h=subt(J,s);
    H=[H;{([char(h)])});
    dH=eval(det(h));
    H=[H;{(['|F`|',num2str(s),')| =
',num2str(dH)])});
    if dH<=-1|dH>=1
        m=1;c=0;
        while m>e
            h1={(['Karena |F`| = ',num2str(m),'
> ',num2str(e),...
                ' = e, maka iterasi lanjut'])});
            h2={([' ']);
            h3={(['Iterasi ke-',num2str(c+1)]});
            c=c+1;
            FX=eval(subt(F,s));
            X=eval(subt(J,s));

```




```

a=inv(X)*FX;
s=s-a';
h4={(['F = ', num2str(s)])};
m=max(abs(FX));
if m>e
    h1={(['Karena |F`| =
', num2str(m), ' > ', num2str(e), ...
' = e, maka iterasi lanjut'])};
    h2={([''])};
    h3={(['iterasi ke-
', num2str(c+1)])};
    FX=eval(subt(F,s));
    h4={(['F(', num2str(s), '=
', num2str(FX') ])};
    c=c+1;
    X=eval(subt(J,s));
    a=inv(X)*FX;
    s=s-a';
    h5={(['F = ', num2str(s)])};
    m=max(abs(FX));
end
H=[H;h1;h2;h3;h4;h5];
end
H=[H;{(['karena |F`| = ', num2str(m), '<
', num2str(e), ...
' = e, maka iterasi berhenti'])}];
else
    H=[H;{(['Nilai awal Anda Salah'])}];
end
set(handles.listbox1,'string',H)
end

```

d. Pada tombol **Hapus**, ketikkan *scribs* berikut.

```

function pushbutton2_Callback(hObject, eventdata,
handles)
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.listbox1,'string','');

```

e. Pada tombol **Home**, ketikkan *scribs* berikut.

```

function pushbutton3_Callback(hObject, eventdata,
handles)
h=waitbar(0,'Silahkan Menunggu.....');
for i=1:300
    waitbar(i/300);
end

```



```
close(h);
close(L);
Interface;
```

Tombol Home ini nantinya akan berfungsi pada saat Anda akan membuat Tampilan Utama dalam menggabungkan semua metode dalam satu GUI.

- f. Tekan tombol RUN berwarna hijau paling atas GUI Anda atau bisa juga tekan tombol F5 pada keyboard.

4. Simulasi

Untuk melihat hasil program yang telah jadi. Kita mencoba dengan menyelesaikan suatu masalah numerik. Jalankan program, isikan fungsi non linier pada label "f(x) dan g(x)", interval awal yang digunakan pada label "s", dan nilai error pada label "e". tampilan seperti gambar di bawah ini.

Misalkan Anda akan menentukan solusi dari pertidaksamaan linier berikut:

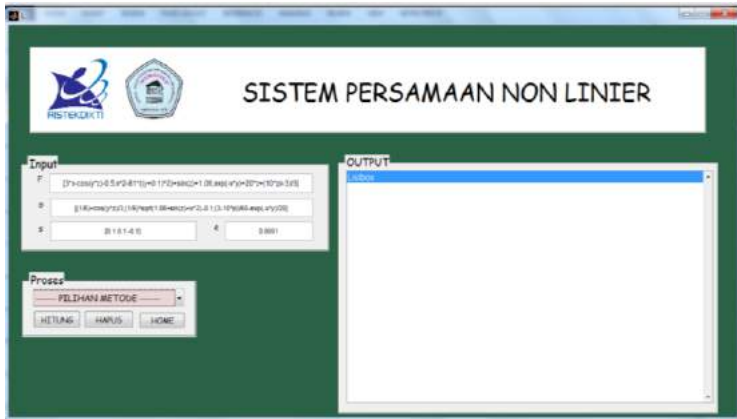
$$F(x, y, z) = \begin{cases} 3x - \cos(yz) - 0.5 \\ x^2 - 81(y + 0.1)^2 + \sin z + 1.06 \\ e^{-xy} + 20z + \frac{10\pi - 3}{3} \end{cases}$$

Dengan fungsi analitik solusi untuk x, y, dan z adalah didefinisikan sebagai fungsi G(x,y,z).

$$G(x, y, z) = \begin{cases} \frac{0.5 + \cos(yz)}{3} \\ \frac{1}{9} \sqrt{1.06 + \sin z + x^2 - 0.1} \\ \frac{3 - 10\pi}{60} - \frac{e^{-xy}}{20} \end{cases}$$

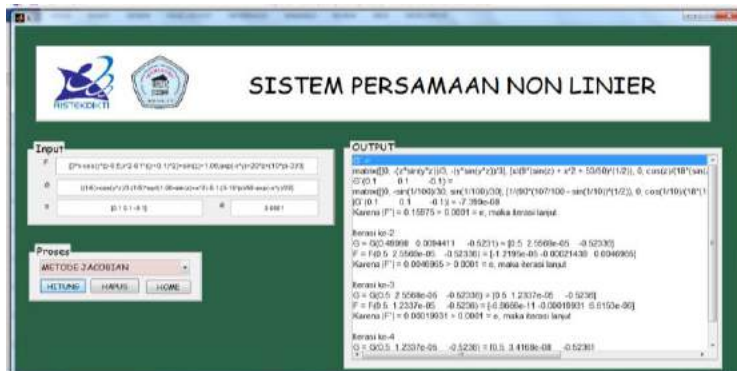
dengan error (e) = 0,0001 pada interval nilai awal [0.1, 0.1, -0.1]





Gambar 2.21 Hasil *Running* GUI SPNL

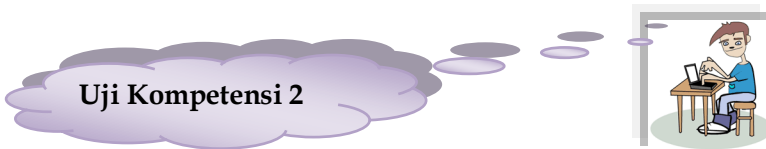
Selanjutnya, pada Panel Proses pilih menu metode yang akan digunakan untuk menyelesaikan permasalahan numerik yang diberikan, misalnya pertama Anda menggunakan **Metode Jacobian**, maka silahkan klik button “Hitung”. Hasil perhitungan langsung ditampilkan pada panel Output di sebelah kanan. Adapun tampilan seperti gambar di bawah ini.



Gambar 2.22 Simulasi Metode Jacobian

Dari GUI tersebut diperoleh nilai $x = 0.5$, $y = 0$ dan $z = -0.52359$ dengan $F(x,y,z) = [0, 0, 0]$.

Untuk metode sistem persamaan non linier yang lain cara menggunakannya tidak jauh berbeda dengan cara di atas, tinggal memilih metode, kemudian tekan tombol “**Hitung**”.

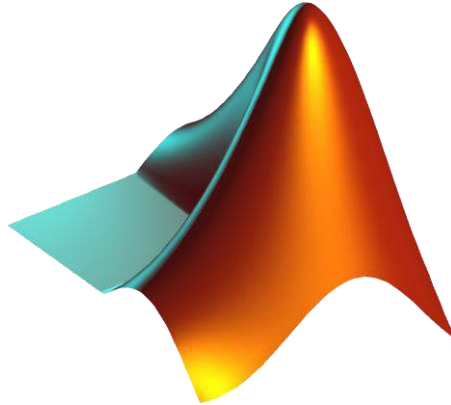


Menggunakan GUI yang sudah Anda buat, selesaikan persoalan berikut ini.

1. Tentukan akar persamaan dari:
 - a. $f(x) = \tan(x-1)$
 - b. $f(x) = 2e^{-x} - \sin x$
 - c. $f(x) = 3x^3 + 4x^2 - 8x - 1$
2. Tentukan solusi untuk SPNL berikut
 - a. $x^2 + y^2 + z^2 = 9; xyz = 1; x + y - z^2 = 0$
 - b. $x^2 + x - y = 1; y - \sin x^2 = 0$
 - c. $x^3 + 3y^2 = 21; x^2 + 2y = 0$
 - d. $xyz - x^2 + y^2 = 1.34; xy - z^2 = 0.009; e^x - e^y + z = 0.41$







BAB 3

RANCANG BANGUN APLIKASI PROJECT

Tujuan Perkuliahan:

1. Mahasiswa mampu membuat *MainMenu* untuk menggabungkan program aplikasi.
2. Mahasiswa mampu merancangbangunkan Aplikasi Project untuk GUI PNL dan SPNL

3.1	Membuat <i>MainMenu</i> (Menu Utama)
-----	--------------------------------------

Pada Bab 2 sudah dibangun GUI untuk simulasi dan solusi numerik Persamaan Non Linier (PNL) dan Sistem Persamaan Non Linier (SPNL), sehingga diperoleh pembagian metode-metode sesuai Tabel 3.1 berikut ini.

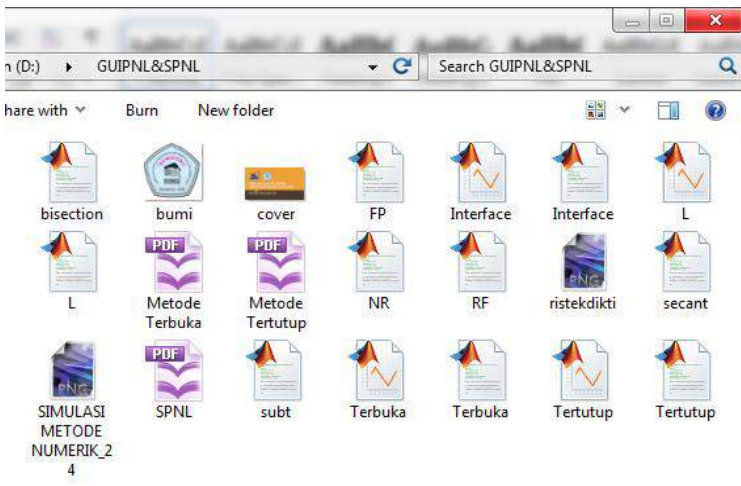


Tabel 3.1 Pembagian MainMenu

Kategori	Metode
PNL	Metode Tertutup <i>a. Bisection</i> <i>b. Regula Falsi</i>
	Metode Terbuka <i>a. Secant</i> <i>b. Newton Raphson</i> <i>c. Fixet Point</i>
SPNL	<i>Jacobian</i> <i>Gauss Seidel</i> <i>Newton Raphson</i>

Sebelumnya, Anda harus memperhatikan dan menyiapkan beberapa file pendukung seperti:

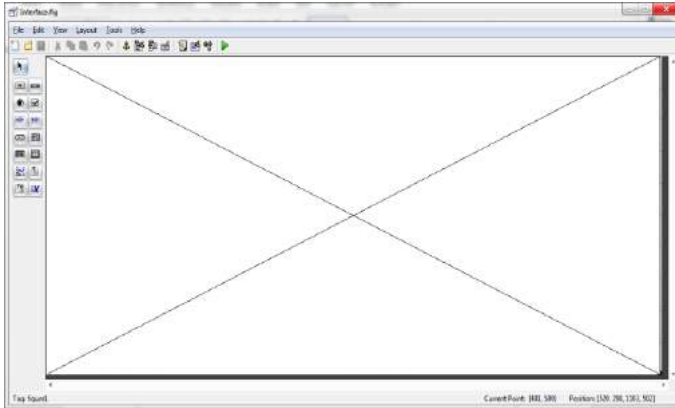
1. Intisari semua metode baik PNL dan SPNL, file tipe PDF.
2. 2 buah logo, file tipe JPEG. Logo ini sifatnya tidak wajib, karena hanya berfungsi untuk memperindah GUI.
3. Menyatukan semua file m-function, PDF, dan JPEG dalam satu folder.



Gambar 3.1 Kumpulan File Pendukung



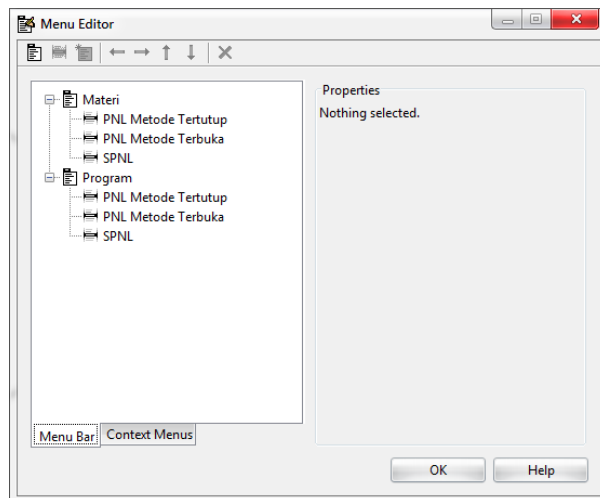
Selanjutnya, perhatikan tampilan GUI awal berikut (sesuaikan atribut yang digunakan). Berilah nama GUI Anda dengan "Interface".



Gambar 3.2 Tampilan Awal GUI MainMenu

Kemudian ikuti langkah-langkah berikut ini.

1. Klik *TollBar Menu Editor* → *New Menu*. Buatlah Menu sesuai Tabel 3.1 di atas, sehingga diperoleh hasil sesuai Gambar 3.2 di bawah ini.



Gambar 3.3 Menu-menu Pada MainMenu



2. Pada masing-masing menu, silahkan Anda klik, kemudian klik tombol *View (Callbacks)* dan ketikkan *scribs* berikut.

Menu	SubMenu	<i>Scribs (Callbacks)</i>
Materi	PNL Metode Tertutup	<code>function</code> Untitled_2_Callback(hObject, eventdata, handles) <code>open('Metode Tertutup.pdf')</code>
	PNL Metode Terbuka	<code>function</code> Untitled_4_Callback(hObject, eventdata, handles) <code>open('Metode Terbuka.pdf')</code>
	SPNL	<code>function</code> Untitled_5_Callback(hObject, eventdata, handles) <code>open('SPNL.pdf')</code>
Program	PNL Metode Tertutup	<code>function</code> Untitled_7_Callback(hObject, eventdata, handles) Tertutup
	PNL Metode Terbuka	<code>function</code> Untitled_8_Callback(hObject, eventdata, handles) Terbuka
	SPNL	<code>function</code> Untitled_9_Callback(hObject, eventdata, handles) L

3. Pada *Axes1*, klik kanan → *Callbacks*, ketikkan *scribs* berikut:
`function Axes1_CreateFcn(hObject, eventdata, handles)`
`imshow('cover.jpg')`
4. Tekan tombol *Running*. Maka akan muncul GUI *MainMenu* seperti berikut.





Gambar 3.4 Tampilan Hasil *Running MainMenu*

5. Lakukan simulasi sesuai tombol yang ada, maka akan muncul kotak dialog seperti GUI yang sudah dibahas pada Bab 2 sebelumnya.

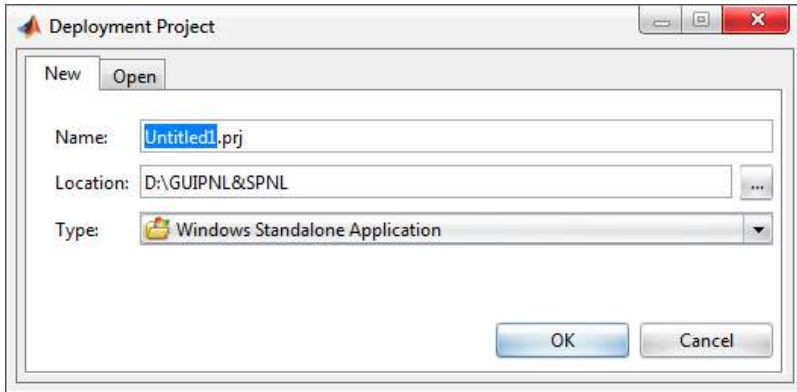
3.2 Membuat Aplikasi Project

Aplikasi *project* sangat penting dibuat mengingat file software matlab yang sangat besar. Membuat aplikasi project dibutuhkan supaya aplikasi yang sudah kita buat bias *running* meskipun master (*setup*) matlab tidak ter-instal di PC/Laptop Anda.

Langkah-langkah Kerja:

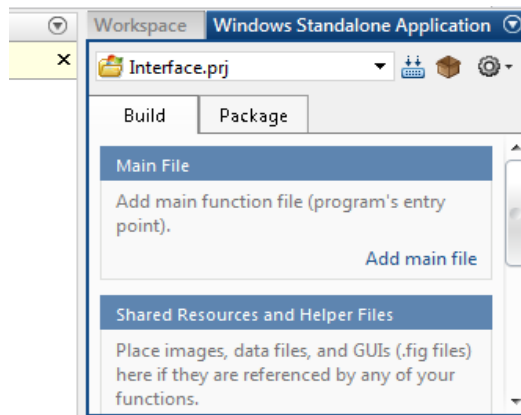
1. Pastikan Anda masih menyimpan file *MainMenu* pada kegiatan sebelumnya.
2. Buka Matlab. Di *Command Windows* silahkan ketik "*deploy-tool*", maka akan muncul kotak dialog seperti berikut.





Gambar 3.5 Tampilan Utama Deploytool

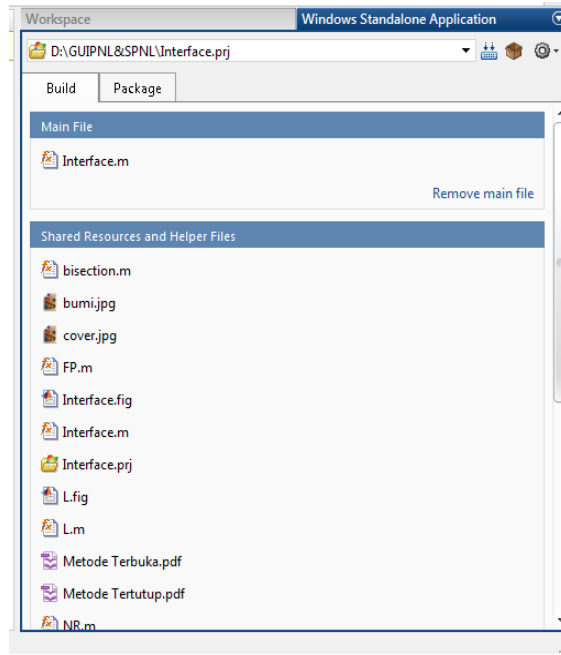
3. Silahkan ganti Name **Untitled1** menjadi *interface* dan klik **OK**. Maka akan muncul windows. Biasanya di sebelah kanan *Command Windows*.



Gambar 3.6 Proses Awal Packing Project

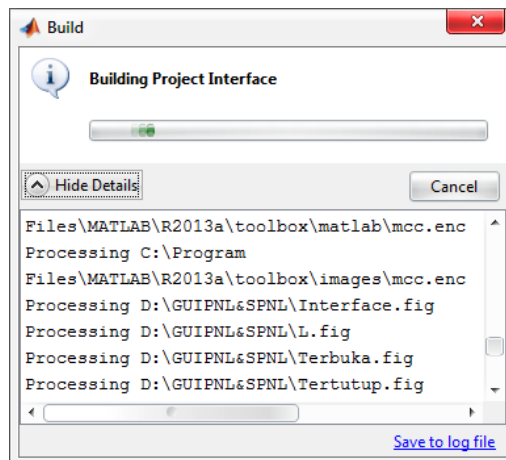
4. Pilih **Build**. Kemudian "*Add main file*". Silahkan pilih "*Interface*", sedangkan pada "*Add files/directories*", blok semua file pendukung kemudian klik **Open**. Maka akan terlihat seperti berikut.





Gambar 3.7 Kumpulan File Pendukung

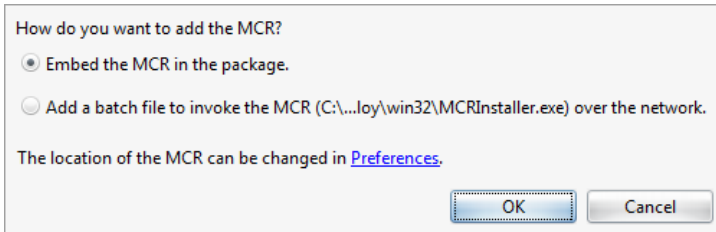
5. Kunci *Interface.m*. kemudian klik Icon *Build* di pojok kanan atas. Silahkan tunggu sampai tulisan "*Cancel*" berubah menjadi "*Close*"



Gambar 3.8 Proses Build

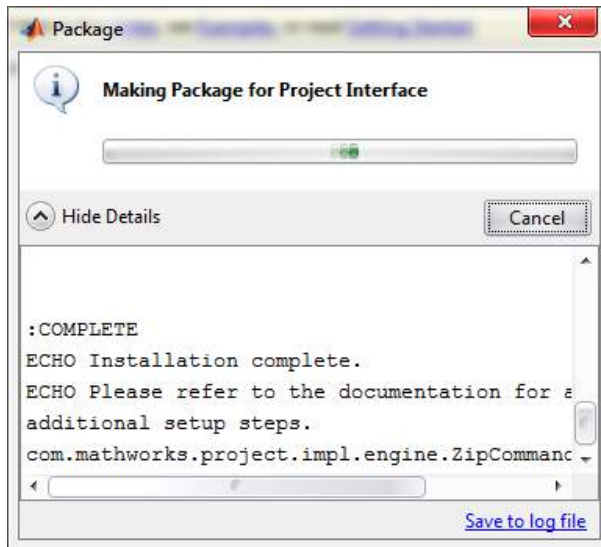


- Selanjutnya, klik tombol *Setting* (bergerigi). Pilih "*Add MCR*". Maka akan muncul kotak dialog. Klik OK.



Gambar 3.9 Proses Setting MCR

- Kunci "*MCRInstaller.exe*", kemudian klik *Icon Package* di pojok kanan atas. Silahkan simpan file*.exe Anda dengan tekan Simpan. Maka akan muncul kotak dialog *Package*. Silahkan tunggu tulisan "*Cancel*" menjadi "*Close*".



Gambar 3.10 Proses Packing Project

- Silahkan tutup program Matlab Anda. Kemudian buka file *Interface_pkg*. Double klik maka akan muncul kotak dialog *Install*. Ikuti saja perintahnya seperti meng-install Matlab.



9. Silahkan cek file tempat menyimpan project tadi, maka akan muncul file baru yang ber-type *Interface.*exe*. Silahkan double klik (ingat: tutup Matlab-nya dulu).
10. Selamat Anda sudah bisa membuat program aplikasi project yang mampu berdiri sendiri tanpa meng-instal Matlab (software induk).





DAFTAR PUSTAKA

- Al-Jawary, M.A. 2017. A Semi-Analytical Iterative Method for Solving Nonlinear thin Film Flow Problems. *Chaos Solitions & Fractals*. 99: 52-56.
- Chen, X. D., Zhang Y. dan Wang, Y. 2016. An Efficient Method Based on Progressive Interpolation for Solving Non-Linear Equations. *Applied Mathematics Letters*. 2016: 67-72.
- Gupta K. D., dan Srivastava, S. 2015. An Iterative Method for Solving General Restricted Linear Equations. *Applied Mathematics and Computation*. 262: 344-353.
- Javidi, M., dan Golbabai A. 2007. Newton-Like Iterative Methods for Solving System of Non-Linear Equations. *Applied Mathematics and Computation*. 192 (2): 546-551.
- Noor, A. M., Waseem, M., dan Noor, I. K. 2015. New Iterative technique for Solving a Siystem of Nonlinear Equations. *Applied Mathematics and Computation*. 271: 446-466.
- Kryshchuk, M. dan Lavendels, J. 2017. Interative Method for Solving a System of Linier Equations. *Procedia Computer Science*. 104: 133-137.
- Perez, O., Amaya, I., dan Correa, R. 2013. Numerical Solution of Certain Exponential and Non-Linier Diophantine Systems of Ewuation by Using a Discrete Particle Swarm Optimization Algorithm. *Applied Mathematics and Computation*. 225: 737-746.
- Ripai. *Pengantar Analisis dan Komputasi Metode Numerik*. Kabupaten Banjarnegara: Sukses Mandiri Press. 2012.
- Syahrudin, "Pengembangan Software Solutif Matematika Aljabar SMP Berbasis Solutif", *Jurnal Paedagoria*, 2015.



Zhang, J. dan Dai H. 2017. A New Iterative method for solving complex symmetric linear systems. *Applied Mathematics and Computation*. 302: 9-20.



GLOSARIUM

Algoritma	:	Metode efektif diekspresikan sebagai rangkaian terbatas dari instruksi-instruksi yang telah didefinisikan dengan baik meliputi proses penghitungan, pemrosesan data, dan penalaran otomatis.
Aplikasi	:	Lebih dikenal dalam perangkat lunak mengarah pada suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.
<i>Bisection</i>	:	Pencarian akar yang membagi dua selang, lalu memilih bagian selang yang berisi akar seharusnya berada untuk diproses lebih lanjut
<i>CallBack</i>	:	Salah satu function dalam bahasa Matlab yang dibuat dan dieksekusi di dalam function lain (function yang memanggilnya)
<i>Command Windows</i>	:	Salah satu tampilan utama atau fitur Matlab sebagai tempat menulis <i>scribs</i> secara langsung untuk menentukan simulasi persoalan matematika
<i>Current Folder</i>	:	Salah satu fitur Matlab tempat link data atau file LocalDisk ke Matlab untuk mempercepat proses <i>running</i> .
<i>Error</i>	:	Disebut juga galat, yakni taraf kesalahan suatu proses iterasi dalam menentukan solusi numerik.
GUI	:	Antarmuka Pengguna Grafis atau APG adalah jenis antarmuka pengguna yang menggunakan metode interaksi pada peranti elektronik secara grafis antara pengguna dan komputer
<i>Interface</i>	:	Batas yang dibagi di mana dua atau lebih komponen terpisah dari <i>informasi</i> pertukaran sistem komputer
Iterasi	:	Sifat tertentu dari algoritme atau program komputer di mana suatu urutan atau lebih dari langkah algoritmik dilakukan di loop program
<i>M-File</i>	:	Sederetan perintah matlab yang dituliskan secara berurutan sebagai sebuah file. hasil dari file yang dibuat akan memiliki ekstensi m yang menandakan bahwa file yang dibuat adalah file MATLAB
Non Linier	:	Sistem yang tidak linier yakni sistem yang tidak memenuhi prinsip superposisi atau sembarang soal dimana peubah yang disolusi tidak dapat ditulis sebagai jumlah linier komponen-komponen tak gayut.
Numerik	:	Teknik-teknik yang digunakan untuk dapat memformulasikan masalah matematis supaya dapat dipecahkan dengan operasi perhitungan



Pemrograman	:	Proses menulis, menguji dan memperbaiki, dan memelihara kode yang membangun suatu program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman.
Persamaan	:	Suatu pernyataan matematika dalam bentuk simbol yang menyatakan bahwa dua hal adalah persis sama
Project	:	Merupakan bentuk compiler dari program aplikasi yang sudah dirunning dan dompiler supaya memungkinkan dioperasikan di perangkat lain
Run	:	Proses melakukan simulasi setelah membangun <i>scribs</i> atau GUI guna melakukan simulasi data
<i>Scribs</i>	:	Bahasa pemrograman Matlab yang disusun secara sistematis berdasarkan algoritma yang diberikan.
Simulasi	:	Suatu proses peniruan dari sesuatu yang nyata beserta keadaan sekelilingnya, dalam Matlab simulasi diartikan sebagai proses uji coba program aplikasi yang sudah dirancang untuk menyelesaikan persoalan yang diberikan.
Sintaksis	:	Ilmu mengenai prinsip dan peraturan untuk membuat kalimat dalam bahasa alami
<i>Software</i>	:	Perangkat lunak atau peranti lunak adalah istilah khusus untuk data yang <i>diformat</i> , dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer.
Solusi	:	Akar persamaan dan sistem persamaan non linier, termasuk juga fungsi jika diketahui titik-titik yang melalui sebuah persamaan.
Struktur	:	Tata ukur atau pedoman yang diklasifi dalam Matlab menjadi tiga besar yakni urut, percabangan, dan perulangan
<i>Swich Case</i>	:	Salah satu struktur percabangan dari bahasa pemograman Matlab
<i>XLabel</i>	:	Untuk menambah label pada garis sumbu X pada grafik dapat menggunakan syntax.
<i>YLabel</i>	:	Untuk menambah label pada garis sumbu Y pada grafik dapat menggunakan syntax.



INDEKS

A

Algoritma · 8, 16, 22, 25, 28, 81

B

Bisection · 6, 7, 8, 12, 16, 19, 31, 42, 43,
48, 49, 70, 81

E

Editor · iv, 3, 37, 40, 43, 71

Enter · 34

F

Fixet · v, 2, 22, 28, 29, 31, 51, 54, 59, 70

Function · 19, 24, 29, 43, 44, 51, 53, 54,
61

G

Guide · 33, 34, 35, 36

K

Keyboard · 36

N

Newton · v, 22, 25, 26, 31, 32, 51, 53, 58,
59, 60, 70, 79

R

Raphson · v, 22, 25, 26, 31, 32, 51, 53,
58, 59, 60, 70





BIODATA PENULIS



Habib Ratu Perwira Negara. Lahir di Mataram tanggal 27 Desember 1988. Pendidikan Dasar di SD Negeri 3 Mataram, kemudian melanjutkan Pendidikan Menengah di SMP Negeri 6 dan SMA Negeri 2 Mataram. Merupakan alumni tahun 2011 di Program Studi Pendidikan Matematika UIN Mataram dan alumni tahun 2014 dari Universitas Sebelas Maret Surakarta. Keseharian aktif mengajar di STMIK Bumigora Mataram dengan mata kuliah yang diampuh Kalkulus, Matematika Diskrit, Aljabar Linier, Metode Numerik, Teknik Operasi Riset, dan Statistika. Penulis aktif menulis di beberapa jurnal baik nasional maupun internasional, termasuk juga aktif melakukan penelitian dan pengabdian yang didanai oleh Kemenristekdikti.



Syaharuddin. Lahir di Empang-Sumbawa pada tanggal 1 Januari 1988. Pendidikan Dasar di SDN Labuhan Aji Tarano, kemudian Pendidikan Menengah di SMPN 1 Tarano dan SMAN 1 Empang. Selanjutnya melanjutkan studi di S1 Pendidikan Matematika UIN Mataram (2007-2011), setelah itu melanjutkan pendidikan magister di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Pernah menjadi Staf Pengajar Matematika di Ganesha Operation (GO) Mataram (2014-2017), sekarang aktif mengajar di Program Studi Pendidikan Matematika UM Mataram. Pernah menulis buku (1) *Delphi for Education: Matematika, Multimedia & Database*, (2) *Matematika Dasar Menggunakan Matlab & Maple*, (3) *5 Software Solusi & Simulasi*



Matematika (*Geogebra, Delphi Math, Maple, WM for Windows, dan Matlab*). Sekarang adalah buku keempatnya.



Kiki Riska Ayu Kurniawati. Lahir di Kota Madiun tanggal 27 Juni 1989. Pendidikan dasarnya ditempuh di SDN Pandean 01. pendidikan menengah pertama ditempuh di SMPN 5 Madiun, selanjutnya menempuh pendidikan menengah atas di SMAN 3 Madiun. Gelar Sarjana Pendidikan Matematika diperolehnya di IKIP PGRI Madiun, setelah itu melanjutkan pendidikan magister di Universitas Sebelas Maret Surakarta. Sekarang menjadi Dosen Negeri di Universitas Islam Negeri Mataram, pada Prodi Pendidikan Matematika. Penulis aktif menulis di beberapa jurnal baik nasional maupun internasional, termasuk juga aktif melakukan penelitian dan pengabdian yang didanai oleh Lembaga UIN Mataram maupun Kementerian Agama RI



Habibi Ratu Perwira Negara, lahir di Mataram 27 Desember 1988. Pendidikan dasar di SD Negeri 3 Mataram, selanjutnya menempuh pendidikan di SMP Negeri 6 Mataram dan pendidikan menengah Atas di SMA Negeri 2 Mataram. Sudah memiliki ketertarikan tersendiri di bidang Matematika sejak kecilnya, hal ini dibuktikan dengan menempuh Sarjana pada bidang Pendidikan Matematika di Universitas Islam Negeri Mataram dan melanjutkan pendidikan Magister di Universitas Sebelas Maret Surakarta. Setelah memperoleh Gelar Magister pendidikan Matematika penulis diangkat menjadi Dosen Negeri di Universitas Islam Negeri Mataram. Sekarang ini penulis sedang menempuh pendidikan S3 pada Prodi Pendidikan Matematika di Universitas Pendidikan Indonesia.







BUATBUKU.COM

----- BUAT AJA DULU -----