



**PENYELESAIAN NUMERIS MASALAH NILAI BATAS
PADA PERSAMAAN DIFERENSIAL BIASA ORDE DUA
BERBASIS KOMPUTASI**
(Studi Komparatif antara Metode Beda Hingga dan Metode Tembakan)



SKRIPSI

 Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Sunan Kalijaga Yogyakarta
untuk Memenuhi Sebagian Syarat Memperoleh
Gelar Sarjana Strata Satu Pendidikan Sains

Oleh: 

M. Kamari
03430368

**PENDIDIKAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA
2008**

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : M. Kamari

NIM : 03430368

Prodi : Pendidikan Matematika

Fakultas : Sains dan Teknologi

menyatakan bahwa skripsi yang berjudul "*Penyelesaian Numeris Masalah Nilai Batas Pada Persamaan Diferensial Biasa Orde Dua Berbasis Komputasi (Studi Komparatif antara Metode Beda Hingga dan Metode Tembakan)*" adalah hasil karya saya sendiri.

Sepanjang pengetahuan saya, karya ini tidak berisi materi yang ditulis oleh orang lain sebagai penyelesaian studi di Universitas Islam Negeri Sunan Kalijaga atau perguruan tinggi lain, kecuali bagian-bagian tertentu yang saya ambil sebagai acuan dengan mengikuti tata cara dan etika penulisan karya ilmiah yang lazim.

Apabila ternyata terbukti bahwa pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 4 Desember 2007

Yang menyatakan,



M. Kamari



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Skripsi Saudara M. Kamari

Lamp. :

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : M. Kamari

NIM : 03430368

Judul Skripsi : Penyelesaian Numeris Masalah Nilai Batas pada Persamaan Diferensial
Biasa Orde Dua Berbasis Komputasi (Studi Komparatif antara Metode
Beda Hingga dan Metode Tembakan)

sudah dapat diajukan kembali kepada Fakultas Sains dan Teknologi Program Studi Pendidikan Matematika UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Pendidikan Sains.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqosyahkan. Atas perhatiannya kami ucapkan terima kasih.

Waslamu'alaikum wr. wb.

Yogyakarta, 4 Desember 2007

Pembimbing I

M. Abrori, S.Si., M.Kom.
NIP. 150293247



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Skripsi Saudara M. Kamari

Lamp. :

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : M. Kamari

NIM : 03430368

Judul Skripsi : Penyelesaian Numeris Masalah Nilai Batas pada Persamaan Diferensial Biasa Orde Dua Berbasis Komputasi (Studi Komparatif antara Metode Beda Hingga dan Metode Tembakan)

sudah dapat diajukan kembali kepada Fakultas Sains dan Teknologi Program Studi Pendidikan Matematika UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Pendidikan Sains.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wasslamu'alaikum wr. wb.

Yogyakarta, 4 Desember 2007

Pembimbing II

Sugiyanto, M.Si.



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/DST/PP.01.1/117/2008

Skripsi/Tugas Akhir dengan judul : PENYELESAIAN NUMERIS MASALAH NILAI BATAS PADA PERSAMAAN DIFERENSIAL BIASA ORDE DUA BERBASIS KOMPUTASI (STUDI KOMPARATIF ANTARA METODE BEDA HINGGA DAN METODE TEMBAKAN)

Yang dipersiapkan dan disusun oleh:

Nama : M. Kamari

NIM : 03430368

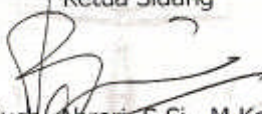
Telah dimunaqasyahkan pada : 18 Januari 2008

Nilai Munaqasyah : A-


dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga.

TIM MUNAQASYAH :

Ketua Sidang


Muchlis Abrori, S.Si., M.Kom.
NIP. 150293247

Penguji I



Fitriyana Yuli S., M.Si.

Penguji II



Yudi Ari Adi, M.Si.

Yogyakarta, 24 Januari 2008

UIN Sunan Kalijaga

Fakultas Sains dan Teknologi

Dekan




Dra. Maizer Said Nahdi, M.Si.

NIP. 159219153

MOTTO

*“ Hariku adalah hari ini,
bukan lusa atau esok
yang belum tentu matahari bersinar di dalamnya. ”*

Laa Tahzan ‘Aidh al-Qorn

PERSEMBAHAN

Karya ini kusembahkan kepada
Kampus Tercinta
Universitas Islam Negeri Sunan Kalijaga Yogyakarta

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “*Penyelesaian Numeris Masalah Nilai Batas pada Persamaan Diferensial Orde Dua Berbasis Komputasi (Studi Komparatif antara Metode Beda Hingga dan Metode Temabakan)*.” Shalawat serta salam semoga senantiasa tercurahkan kepada Rasulullah SAW sebagai *revolusioner* Islam dan *uswatun khasanah* bagi seluruh umat dan rahmat bagi seluruh alam.

Penulis menyadari bahwa terselesaikannya skripsi ini tidak terlepas dari peran serta dari berbagai pihak yang telah memberikan dukungan baik yang bersifat materiil maupun moril. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ibu Dra. Hj. Maizer Said Nahdi, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
2. Ibu Dra. Khurul Wardati, M.Si. selaku Ketua Program Studi Pendidikan Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Muchammad Abrori, S.Si., M.Kom. selaku dosen pembimbing akademik sekaligus dosen pembimbing I yang senantiasa dengan penuh kesungguhan dan keikhlasan memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
4. Bapak Sugiyanto, M.Si. selaku dosen pembimbing II yang dengan penuh keikhlasan memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.

5. Bunda dan Kakak sekeluarga tercinta,”*Jazakumullah bi ahsanil jaza* atas segala usaha dan peluh keringatnya selama ini, dan semoga dengan izin Allah, saya bisa memberikan yang terbaik untuk keluarga dan Islam.”
6. Keluarga Pak Sri, keluarga Bu Wid, keluarga Ust. Zein (alm) atas dukungan dan bantuannya selama ini. *Jazakumullah khairan katsiran*.
7. Sahabat kecilku, Evi, Nabila, Iqoh, Iil, yang telah menemani penulis dengan riang dan canda tawa.
8. *Asatidz* dan *Ikhwah* di PP. Fauzul Muslimin Kotagede dan TPM’03 yang telah berbagi ilmunya dan menemani perjalanan serta memberi warna kehidupan penulis selama di Jogja.
9. Serta berbagai pihak yang tidak bisa penulis sebutkan satu persatu.

Atas semua kebaikan ini, semoga Allah SWT menerimanya sebagai amal kebaikan dan mendapatkan balasan yang lebih baik di sisi-Nya.

Harapan penulis mudah-mudahan skripsi ini bermanfaat bagi penulis pada khususnya dan pembaca pada umumnya.

Yogyakarta, 3 Desember 2007

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
SURAT PERNYATAAN KEASLIAN	ii
HALAMAN NOTA DINAS PEMBIMBING I	iii
HALAMAN NOTA DINAS PEMBIMBING II	iv
HALAMAN PENGESAHAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xviii
ABSTRAK	xix
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	7
C. Batasan Masalah	8
D. Rumusan Masalah	8
E. Tujuan dan Kegunaan Penelitian	9
F. Sistematika Pembahasan	10

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	12
A. Tinjauan Pustaka	12
B. Landasan Teori	13
1. Persamaan diferensial	13
2. Persamaan diferensial biasa	16
3. Penyelesaian persamaan diferensial biasa	21
4. Diferensiasi numerik (<i>Numerical Differentiation</i>)	22
5. Metode dekomposisi LU	23
6. Masalah nilai awal pada persamaan diferensial biasa	25
7. Metode Euler	27
8. Metode Runge Kutta	28
9. Metode Runge Kutta orde empat	30
10. Masalah nilai batas pada persamaan diferensial biasa	33
11. Metode beda hingga (<i>finite difference method</i>)	34
12. Metode tembakan (<i>shooting method</i>)	39
13. Pengantar perangkat lunak MATLAB	41
 BAB III METODE PENELITIAN	 42
A. Jenis Penelitian	42
B. Sumber Data	42
C. Metode Pengumpulan Data dan Analisis Data	43
D. Perangkat Pendukung	44

E. Algoritma dan Flowchart Metode Beda Hingga	44
1. Algoritma metode beda hingga	44
2. Flowchart metode beda hingga	46
F. Algoritma dan Flowchart Metode Tembakan	49
1. Algoritma metode tembakan	49
2. Flowchart metode tembakan	52
G. Contoh Soal dan Penyelesaian Masalah Nilai Batas (MNB)	54
Soal 3.1	54
a. Metode analitis	54
b. Metode beda hingga	57
c. Metode tembakan	67
d. Perbandingan hasil perhitungan MNB	104
Soal 3.2	105
a. Metode analitis	105
b. Metode beda hingga	106
c. Metode tembakan	110
d. Perbandingan hasil perhitungan MNB	121
Soal 3.3	122
a. Metode analitis	122
b. Metode beda hingga	124
c. Metode tembakan	127
d. Perbandingan hasil perhitungan MNB	139

BAB IV PENERAPAN DAN PEMBAHASAN	140
A. Penerapan	140
1. Menu utama 1(PROGRAM_APLIKASI)	141
2. Menu utama 2 (METODE_BTM)	144
a. Kotak input	145
b. Kotak output	146
c. Menu input	147
d. Menu help	149
e. Menu output	151
f. Simpan nilai	152
g. Cetak nilai	153
h. Menu keluar	153
B. Pembahasan	154
1. Cara menjalankan proses penyelesaian numeris MNB dengan metode beda hingga dan metode tembakan menggunakan program MATLAB	154
Soal 4.1	154
Soal 4.2	159
Soal 4.3	164
2. Keluaran (<i>output</i>) Program	170

BAB V PENUTUP	171
A. Kesimpulan	172
B. Saran-Saran	173
DAFTAR PUSTAKA	174
LAMPIRAN-LAMPIRAN	176

DAFTAR TABEL

Tabel 3.1 Hasil perhitungan MNB soal 3.1 secara analitis	57
Tabel 3.2. Hasil perhitungan MNB soal 3.1 dengan metode Beda Hingga	67
Tabel 3.3. Nilai-nilai u dan v metode tembakan pada soal 3.1	101
Tabel 3.4. Nilai-nilai w_n metode Tembakan pada soal 3.1	102
Tabel 3.5. Hasil perhitungan MNB soal 3.1 dengan metode Tembakan	103
Tabel 3.6. Kesalahan mutlak dan kesalahan relatif MNB metode Beda Hingga dan metode Tembakan terhadap nilai eksak pada soal 3.1	104
Tabel 3.7. Hasil perhitungan MNB soal 3.2 dengan metode Beda Hingga	109
Tabel 3.8. Nilai-nilai u dan v metode tembakan pada soal 2.3	120
Tabel 3.9. Nilai-nilai w_n metode tembakan pada soal 2.3	120
Tabel 3.10. Hasil perhitungan MNB soal 2.3 dengan metode Tembakan	121
Tabel 3.11. Perbandingan Hasil Perhitungan MNB metode Beda Hingga dan metode Tembakan soal 2.3	121
Tabel 3.12. Hasil perhitungan MNB soal 3.3 secara analitis	123
Tabel 3.13. Hasil perhitungan MNB soal 3.3 dengan metode Beda Hingga	127
Tabel.3.14. Nilai-nilai u dan v metode tembakan pada soal 3.3	138
Tabel 3.15. Nilai-nilai w_n metode tembakan pada soal 3.3	138
Tabel 3.16. Hasil perhitungan MNB soal 3.3 dengan metode Tembakan	138
Tabel 3.17. Perbandingan Hasil Perhitungan MNB metode Beda Hingga dan metode Tembakan soal 3.3	139
Tabel 3.18. Kesalahan mutlak dan kesalahan relatif MNB metode Beda Hingga dan metode Tembakan terhadap nilai eksak pada soal 3.3	139

DAFTAR GAMBAR

Gambar 4.1. <i>Tampilan Menu Utama MATLAB Common Window</i>	140
Gambar 4.2. <i>MATLAB Common Window dengan File Program_Aplikasi</i>	141
Gambar 4.3. <i>Menu Utama Program Aplikasi</i>	142
Gambar 4.4. <i>Jendela Profil</i>	142
Gambar 4.5. <i>Jendela Petunjuk</i>	143
Gambar 4.6. <i>Jendela Keluar</i>	143
Gambar 4.7. <i>Program Beda Hingga dan Tembakan Berbasis MATLAB</i>	144
Gambar 4.8. <i>Jendela Hasil Penyelesaian</i>	146
Gambar 4.9. <i>Jendela Simpan Input</i>	147
Gambar 4.10. <i>Jendela Buka Input</i>	147
Gambar 4.11. <i>Jendela figure METODE BTM setelah direset</i>	148
Gambar 4.12. <i>Jendela Help</i>	149
Gambar 4.13. <i>Jendela Ketentuan Input Koefisien</i>	150
Gambar 4.14. <i>Jendela tentang Output</i>	151
Gambar 4.15. <i>Proses Penyelesaian MNB</i>	151
Gambar 4.16. <i>Jendela Simpan Nilai</i>	152
Gambar 4.17. <i>Jendela Cetak Nilai</i>	153
Gambar 4.18. <i>Jendela Keluar</i>	153
Gambar 4.19. <i>Figur Soal 4.1</i>	156
Gambar 4.20. <i>Grafik Soal 4.1</i>	156
Gambar 4.21. <i>Hasil Penyelesaian Soal 4.1 berbasis MATLAB (Modifikasi)</i>	157

Gambar 4.22. <i>Figur Soal 4.2</i>	160
Gambar 4.23. <i>Grafik Soal 4.2</i>	161
Gambar 4.24. <i>Hasil Penyelesaian Soal 4.2 berbasis MATLAB</i>	162
Gambar 4.25. <i>Figur Soal 4.3</i>	165
Gambar 4.26. <i>Grafik Soal 4.3</i>	166
Gambar 4.27. <i>Hasil Penyelesaian Soal 4.3 berbasis MATLAB</i>	167

DAFTAR LAMPIRAN

1. Listing Program Aplikasi 176
2. Listing Metode BTM 178

**PENYELESAIAN NUMERIS MASALAH NILAI BATAS
PADA PERSAMAAN DIFERENSIAL BIASA ORDE DUA
BERBASIS KOMPUTASI
(Studi Komparatif antara Metode Beda Hingga dan Metode Tembakan)**

Oleh: M. Kamari

ABSTRAK

Dalam perkembangannya, persamaan diferensial mempunyai peranan yang sangat penting dalam kehidupan (alam). Banyak permasalahan dalam bidang ilmu pengetahuan dan teknik yang dapat diformulasikan ke dalam bentuk persamaan diferensial dalam mencari pemecahannya. Tetapi untuk menyelesaikan persamaan diferensial yang rumit dan besar terkadang sulit diselesaikan secara analitis. Oleh karena itu penggunaan metode numeris dalam hal ini sangat tepat.

Penelitian ini bermaksud memberikan penyelesaian masalah nilai batas (MNB) pada persamaan diferensial orde dua secara numeris dengan menggunakan metode Beda Hingga (*Finite Difference Method*) dan metode Tembakan (*Shooting Method*). Dalam menyelesaikan MNB dengan kedua metode tersebut dilengkapi dengan program aplikasi menggunakan MATLAB R2006a sehingga mempercepat dalam penyelesaiannya.

Hasil penelitian ini menunjukkan bahwa nilai keakuratan *metode tembakan* dalam penyelesaian numeris MNB lebih baik daripada *metode beda hingga* karena mempunyai nilai error lebih kecil terhadap nilai sebenarnya (eksak). Ditinjau dari efisiensi waktu, *metode beda hingga* lebih efisien daripada *metode tembakan* karena waktu yang diperlukan untuk menyelesaikan MNB secara numeris lebih sedikit daripada metode tembakan. Demikian juga penyelesaian berbasis komputasi MATLAB lebih efisien daripada penyelesaian secara manual. Ditinjau dari jumlah iterasi, semakin banyak iterasi yang digunakan maka nilai penyelesaian dengan metode beda hingga dan metode tembakan akan semakin mendekati nilai sebenarnya (eksak) atau dengan kata lain nilai error akan semakin kecil.

Kata kunci: MNB, metode beda hingga, metode tembakan, metode analitis.

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Persamaan diferensial adalah suatu persamaan yang memuat derivatif dengan sekurang-kurangnya satu derivatif dari fungsi yang tidak diketahui (Wardiman, 1981).¹ Secara umum, persamaan diferensial dibagi menjadi dua, yaitu *persamaan diferensial biasa* dan *persamaan diferensial parsial*.² Untuk selanjutnya, persamaan diferensial disingkat dengan PD. PD biasa adalah PD yang hanya mengandung satu variabel bebas. Sedangkan PD parsial adalah PD yang mengandung lebih dari satu variabel bebas. Dalam perkembangannya, persamaan diferensial mempunyai peranan yang sangat penting dalam kehidupan (alam). Banyak permasalahan dalam bidang ilmu pengetahuan dan teknik yang dapat diformulasikan ke dalam bentuk persamaan diferensial dalam mencari pemecahannya.³ Selain itu, persamaan diferensial juga sering digunakan sebagai model matematika dalam berbagai bidang, antara lain: bidang Ekonomi, Fisika, Biologi, Teknik, dan sebagainya.⁴

Sehingga dalam hal ini, persamaan diferensial memegang peranan penting dalam kehidupan terutama bagi pengembangan ilmu pengetahuan dan teknologi (IPTEK).

¹ Wardiman, *Diktat Kuliah Persamaan Diferensial Elementer*, (Yogyakarta: Jurusan FMIPA UGM, 1981).

² Bambang Triatmojo, *Metode Numerik*, (Yogyakarta: Beta Offset, 2002), hal. 165.

³ *Ibid.*, hal. 165.

⁴ Bambang Soedijono dan B. Susana, *Model Matematik (Alat Penyusun Model)*, (Jakarta: Karunia, 1993), hal. 2.

Terdapat dua macam cara dalam menyelesaikan persamaan diferensial⁵. *Pertama*, secara analitis. Cara ini biasanya digunakan untuk menyelesaikan persamaan diferensial yang mempunyai bentuk sederhana. *Kedua*, secara numerik. Cara ini biasanya digunakan dalam menyelesaikan persamaan diferensial yang sulit dikerjakan secara analitis.

Sejak akhir tahun 1940-an, ketersediaan yang meluas dari komputer digital telah menuju kepada suatu ledakan dalam penggunaan dan pengembangan metode numerik.⁶ Metode numerik merupakan alat pemecahan masalah yang sangat ampuh yang dapat digunakan untuk menyelesaikan suatu persamaan (diferensial) yang besar dan rumit, masalah ketidaklinieran, dan geometri yang tidak dapat diselesaikan secara analitis.⁷

Solusi-solusi yang diberikan dalam metode numerik merupakan nilai yang mendekati nilai eksak sehingga nilai-nilai tersebut disebut sebagai nilai hampiran/pendekatan.⁸ Dalam penyelesaiannya, nilai kesalahan numerik harus cukup kecil terhadap tingkat kesalahan yang ditetapkan.⁹

Penyelesaian persamaan diferensial dengan metode numerik ini dibedakan menjadi dua, yaitu: metode numerik untuk penyelesaian masalah nilai awal dan metode numerik untuk penyelesaian masalah nilai batas.

⁵ Bambang Triatmodjo, hal. 1.

⁶ Steven C. Chapra dan Raymond P. Canale, *Metode Numerik* (terjemahan), (Jakarta: Penerbit Erlangga, 1996), hal. 3.

⁷ *Ibid.*, hal. 3.

⁸ Bambang Triatmodjo, *Metode Numerik* (Yogyakarta: Beta Offset, 2002), hal. 1.

⁹ *Ibid.*, hal. 1.

a. Metode Numerik untuk Penyelesaian Masalah Nilai Awal

Penyelesaian masalah nilai awal merupakan penyelesaian persamaan diferensial yang ditinjau dari keadaan awal yang diberikan pada titik tunggal tertentu.¹⁰ Metode-metode yang sering digunakan untuk menyelesaikan masalah nilai awal pada persamaan diferensial biasa yaitu: Metode Euler, Metode Heun, Metode Deret Taylor, dan Metode Runge Kutta.

Pertama, *Metode Euler*, adalah salah satu dari metode satu langkah yang paling sederhana. Jika dibandingkan dengan metode lain, metode ini mempunyai banyak kekurangan dalam penyelesaian persamaan diferensial biasa, antara lain; kesalahan pemotongan dan kesalahan pembulatan.¹¹ Kedua, *Metode Heun*, merupakan modifikasi dari Metode Euler. Modifikasi ini digunakan untuk memperkirakan kemiringan Φ dan dua turunan pada interval, yaitu pada ujung awal dan akhir.¹² Ketiga, *Metode Deret Taylor*, merupakan metode yang dapat digunakan untuk memperbaiki Metode Euler dengan memperhitungkan lebih banyak suku dari deret Taylor sendiri. Kelemahan dari metode ini adalah dalam mencari penyelesaian akan menjadi lebih sulit apabila fungsi sulit diturunkan. Keempat, *Metode Runge Kutta*, merupakan metode yang memberikan ketelitian hasil yang lebih baik dan tidak memerlukan turunan fungsi sebagaimana

¹⁰ Samuel D. Conte dan Carl De Boor, *Dasar-Dasar Analisis Numerik dengan Suatu Pendekatan Algoritma*, (Jakarta: Erlangga, 1993), hal. 366.

¹¹ Bambang Triatmodjo, *Metode Numerik*, (Yogyakarta: Beta Offset, 2002), hal. 173.

¹² *Ibid.*, hal. 176.

pada Deret Taylor. Metode ini dapat digunakan pada persamaan diferensial orde dua, tiga dan empat.¹³ Selain itu juga ada metode lain yang dapat digunakan dalam mencari penyelesaian masalah nilai awal yaitu *Metode Adams – Bashforth – Moulton*.

b. Metode Numerik untuk Penyelesaian Masalah Nilai Batas

Penyelesaian masalah nilai batas merupakan penyelesaian persamaan diferensial yang ditinjau dari keadaan awal yang diberikan pada lebih dari satu titik.¹⁴ Metode-metode yang sering digunakan untuk menyelesaikan masalah nilai batas pada persamaan diferensial biasa adalah:

1) *Metode Beda Hingga*

Untuk menyelesaikan masalah nilai batas dengan metode beda hingga, setiap turunan yang muncul dalam persamaannya maupun dalam persyaratan batasnya diganti dengan suatu aproksimasi diferensiasi yang tepat.¹⁵ Diferensiasi yang sering digunakan adalah diferensi pusat (*central differensi*).

2) *Metode Tembakan*

Metode tembakan merupakan metode yang digunakan untuk menyelesaikan masalah nilai batas dengan menggunakan

¹³ *Ibid.*, hal. 182.

¹⁴ Samuel D. Conte dan Carl De Boor, *Dasar-Dasar Analisis Numerik dengan Suatu Pendekatan Algoritma*, (Jakarta: Erlangga, 1993), hal. 366.

¹⁵ *Ibid.*, hal. 367.

metode iterasi numerik untuk mencapai kemiringan yang sebenarnya.¹⁶

3) Metode Kolokasi

Metode kolokasi merupakan metode penyelesaian masalah nilai batas dengan menggunakan fungsi polinomial sebagai dasar penyelesaiannya.¹⁷

Masalah nilai batas pada persamaan diferensial biasa, pada umumnya dapat diselesaikan secara analitik bukan secara numerik yang langkah akhir penyelesaiannya menggunakan teknik integrasi. Namun demikian, tidak semua nilai batas dapat diselesaikan secara analitik. Dalam hal ini penyelesaian dapat diselesaikan dengan metode numerik.

Pada umumnya, metode numeris tidak mengutamakan diperolehnya jawaban yang tepat (eksak) tetapi lebih pada mengusahakan perumusan metode yang menghasilkan jawaban pendekatan yang berbeda (hampir sama) dengan jawaban eksak sebesar suatu nilai yang dapat diterima berdasarkan pertimbangan yang cukup dapat memberikan penyelesaian pada permasalahan yang dihadapi.¹⁸

Dari uraian di atas, peneliti tertarik untuk meneliti penggunaan Metode Beda Hingga (*Finite Difference Method*) dan Metode Tembakan (*Shooting Method*) dalam pemecahan masalah nilai batas pada persamaan diferensial biasa orde dua yang linier. Dari uraian tentang metode yang digunakan dalam mencari penyelesaian masalah nilai awal, terdapat metode yang mempunyai

¹⁶ *Ibid.*, hal. 371.

¹⁷ *Ibid.*, hal. 375.

¹⁸ Harijono Djojodihardjo, *Metode Numerik*, (Jakarta: Erlangga, 1983), hal. 5.

tingkat ketelitian lebih dibandingkan metode lain. Oleh karena itu, peneliti tertarik untuk mengetahui lebih dalam mengenai efisiensi, kelemahan dan kelebihan metode beda hingga dan metode tembakan kemudian membandingkannya.

Seiring perkembangan teknologi yang semakin pesat, keberadaan komputer dengan berbagai macam bahasa pemrograman yang ditawarkan semakin mempermudah seseorang memecahkan masalah dalam berbagai bidang kehidupan, terutama bidang komputasi. Terdapat beberapa macam bahasa pemrograman yang dapat digunakan dalam menyelesaikan persamaan diferensial, antara lain: BASIC, PASCAL, MATLAB, FORTRAN, MAPLE, C dan sebagainya.

Pada penyelesaian persamaan diferensial secara manual memungkinkan sekali terjadi kesalahan yang lebih sering dilakukan oleh pelaku sehingga dalam penyelesaian persamaan diferensial diperlukan sarana pendukung yang berfungsi sebagai alat penguji kebenaran sehingga dengan ini diharapkan akan memperkecil terjadinya kesalahan. Selain itu juga dengan adanya suatu program akan lebih menghemat waktu.

Sebagai sarana pendukung, peneliti memilih MATLAB sebagai bahasa pemrograman (komputasi) dalam membantu penyelesaian masalah nilai batas dengan kedua metode di atas. Alasan pemilihannya dikarenakan MATLAB merupakan bahasa pemrograman (komputasi) yang dapat digunakan untuk mengerjakan operasi matematika yang memberi kemudahan dalam

mengimplementasikan penggunaan metode numeris.¹⁹ Selain itu, MATLAB juga dilengkapi dengan berbagai fasilitas interaktif, visualisasi, pemodelan, program aplikasi, pembuatan grafis, mendesain algoritma, analisis data, simulasi dan pembuatan prototype.²⁰

B. Identifikasi Masalah

Secara umum, persamaan differensial dibagi menjadi dua, yaitu *persamaan diferensial biasa* dan *persamaan diferensial parsial*.²¹

Persamaan diferensial biasa adalah persamaan diferensial yang hanya mempunyai satu variabel bebas dan derivatif-derivatifnya adalah derivatif biasa. Persamaan diferensial parsial adalah persamaan diferensial yang mempunyai lebih dari satu variabel bebas dan derivatif-derivatifnya adalah derivatif parsial.

Berdasarkan uraian latar belakang di atas, dalam menyelesaikan persamaan diferensial biasa pada masalah nilai batas terdapat beberapa metode yang dapat digunakan, yaitu; metode beda hingga, metode tembakan, dan metode kolokasi.

Selain itu, dalam menyelesaikan kedua PD di atas biasanya digunakan cara biasa secara manual dan analitis (tidak secara numeris). Penyelesaian persamaan diferensial dengan cara manual akan membutuhkan banyak waktu dalam menyelesaikannya. Selain itu juga diperlukan ketekunan dan ketelitian

¹⁹ Talib Hashim Hasan, *Belajar Sendiri Dasar-Dasar Pemrograman MATLAB* (Yogyakarta: Gava Media, 2005), hal. 1.

²⁰ *Ibid.*, hal. 2.

²¹ Bambang Triatmojo, *Metode Numerik* (Yogyakarta: Beta Offset, 2002), hal. 165.

dalam penyelesaiannya. Berdasarkan latar belakang masalah tersebut di atas, metode numerik (seiring perkembangan teknologi) memainkan peranan yang sangat penting dalam penyelesaian persamaan diferensial dengan bantuan komputer (MATLAB).

C. Batasan Masalah

Penelitian dan pengembangan sistem komputasi numerik ini ditekankan pada:

1. Masalah nilai batas pada PD biasa orde dua yang linier dengan koefisien konstan/variabel.
2. Perbandingan nilai error penyelesaian masalah nilai batas dengan menggunakan metode beda hingga dan metode tembakan terhadap nilai sebenarnya (eksak) dan efisiensi waktu yang diperlukan.
3. Penyelesaian masalah nilai batas yang dilengkapi dengan program berbasis MATLAB.

D. Rumusan Masalah

Berdasarkan pemaparan latar belakang masalah, dapat dirumuskan beberapa masalah antara lain:

1. Bagaimanakah perbandingan (kelebihan dan kekurangan) antara metode analitik, metode beda hingga dan metode tembakan dalam penyelesaian MNB?

2. Bagaimanakah nilai kesalahan mutlak dan relatif penyelesaian masalah nilai batas persamaan diferensial orde dua dengan metode beda hingga dan metode tembakan terhadap nilai sebenarnya?
3. Bagaimanakah program komputasi MATLAB dalam penyelesaian masalah nilai batas persamaan diferensial biasa orde dua dengan menggunakan metode analitis, metode beda hingga dan metode tembakan?

E. Tujuan dan Kegunaan Penelitian

1. Tujuan penelitian

Tujuan dari penelitian ini antara lain:

- a. Untuk mengetahui perbandingan (kelebihan dan kekurangan) penggunaan antara metode analitik, metode beda hingga dan metode tembakan dalam penyelesaian MNB baik secara manual maupun berbasis MATLAB.
- b. Untuk mengetahui nilai error dalam penggunaan metode beda hingga dan metode tembakan dalam penyelesaian MNB persamaan diferensial biasa orde dua dan efisiensi waktu yang diperlukan.
- c. Menghasilkan program komputasi berbasis MATLAB dalam penyelesaian Masalah Nilai Batas sehingga mempercepat dan mempermudah proses penyelesaian.

2. Kegunaan penelitian

Kegunaan dari hasil penelitian ini antara lain:

- a. Bagi peneliti
 - 1) Mengetahui lebih mendalam tentang penggunaan Metode Beda Hingga dan Metode Tembakan terutama dalam penyelesaian Masalah Nilai Batas.
 - 2) Menguasai bahasa pemrograman MATLAB dalam penggunaannya terutama dalam membuat penyelesaian Masalah Nilai Batas persamaan diferensial biasa orde dua.
- b. Bagi perkembangan keilmuan
 - 1) Memberikan sumbangan pemikiran khususnya dalam bidang matematika dalam penyelesaian Masalah Nilai Batas persamaan diferensial biasa orde dua.
 - 2) Sebagai bahan referensi mahasiswa dalam bidang matematika pada penyelesaian Masalah Nilai Batas.
 - 3) Memperkaya khasanah keilmuan dalam bidang matematika.

F. Sistematika Pembahasan

Sistematika penulisan penelitian ini, terdiri dari:

Bab I Pendahuluan

Bab ini berisi tentang latar belakang masalah, identifikasi masalah, batasan masalah, perumusan masalah, tujuan dan kegunaan penelitian dan sistematika pembahasan.

Bab II Tinjauan Pustaka dan Landasan Teori

Bab ini terdiri dari tinjauan pustaka dan landasan teori. Tinjauan pustaka berisi hasil-hasil penelitian yang berhubungan dengan penelitian. Sedangkan landasan teori berisi tentang tinjauan umum masalah persamaan diferensial biasa, penyelesaian persamaan diferensial biasa, masalah nilai awal, masalah nilai batas, metode beda hingga dan metode tembakan, serta pengenalan perangkat lunak MATLAB.

Bab III Metode Penelitian

Meliputi jenis penelitian, sumber data, alat penelitian, teknik pengumpulan dan analisis data serta algoritma metode beda hingga dan metode tembakan.

Bab IV Hasil Penerapan dan Pembahasan

Pada bab ini berisi tentang penyelesaian numeris masalah nilai batas pada persamaan diferensial biasa dengan metode beda hingga dan metode tembakan, perbandingan metode beda hingga dan metode tembakan, serta programnya berbasis MATLAB.

Bab V Penutup

Bab ini meliputi dua bagian, yaitu kesimpulan dan saran-saran peneliti terhadap masalah yang berkaitan dengan penelitian yang telah dilakukan.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan contoh penyelesaian dan hasil pembahasan dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan efisiensi waktu
 - a. Metode beda hingga lebih efisien daripada metode tembakan karena algoritma yang digunakan lebih sedikit daripada metode tembakan, sehingga secara tidak langsung waktu yang diperlukan untuk penyelesaian MNB dengan metode beda hingga lebih sedikit daripada metode tembakan.
 - b. Penyelesaian berbasis komputasi MATLAB lebih efisien daripada penyelesaian secara manual.

2. Berdasarkan keakuratan hasil

Metode tembakan mempunyai hasil (nilai) penyelesaian lebih akurat daripada metode beda hingga karena nilai error mutlak dan error relatifnya lebih kecil daripada metode beda hingga.

3. Berdasarkan jumlah iterasi (N)

Semakin banyak jumlah iterasi yang digunakan maka nilai penyelesaian beda hingga dan tembakan akan semakin mendekati nilai sebenarnya (eksak) tetapi waktu yang diperlukan untuk menyelesaikannya semakin banyak.

4. Berdasarkan ada tidaknya penyelesaian

Penyelesaian numeris dapat digunakan untuk menyelesaikan MNB yang tidak dapat diselesaikan secara analitis karena penyelesaian analitis tergantung dari ketepatan dalam penggunaan syarat batas dan nilai batas pada *penyelesaian umum* persamaan diferensialnya.

B. Saran-Saran

Dalam kesempatan ini, penulis ingin menyampaikan beberapa hal terkait dengan penelitian ini, antara lain:

1. Pengembangan dan penyempurnaan penelitian ini dapat dilakukan dengan membandingkan tiga metode yaitu; metode beda hingga, metode tembakan, dan metode kolokasi dengan menggunakan program aplikasi MATLAB.
2. Pengembangan dan penyempurnaan pada metode tembakan dalam penelitian ini dapat dilakukan dengan membandingkan penggunaan metode Runge-Kutta orde 2 dan orde 3, metode Euler, ataupun metode-metode yang sering digunakan dalam penyelesaian masalah nilai awal.

DAFTAR PUSTAKA

- Arhami, Muhammad dan Desiani, Anita. 2005. *Pemrograman MATLAB*. Yogyakarta: Penerbit Andi.
- Basuki A.R.1993. *Persamaan Diferensial*. Yogyakarta: FMIPA IKIP Muhammadiyah.
- Ayres, Frank. 1990. *Persamaan Diferensial dalam Satuan SI metric* (alih bahasa Lily Ratna). Jakarta: Penerbit Erlangga.
- Chapra, Steven C. dan Canale, Raymond P.. 1996. *Metode Numerik* (terjemahan). Jakarta: Penerbit Erlangga.
- Conte, Samuel D. dan Boor, Carl De. 1993. *Dasar-Dasar Analisis Numerik dengan Suatu Pendekatan Algoritma*. Jakarta: Erlangga.
- Djojodihardjo, Harijono.1983. *Metode Numerik*. Jakarta: Erlangga.
- Finizio dan Ladas . 1988. *Persamaan Diferensial Biasa dan Penerapan Modern*. (Terjemahan). Jakarta: Erlangga.
- Hasan, Talib Hashim. 2005. *Belajar Sendiri Dasar-Dasar Pemrograman MATLAB*. Yogyakarta: Gava Media.
- Ibraheem, Kais Ismail dan Hisyam, Anwaruddin. 2003. *Metode Numerik untuk Sains dan Teknik dengan MATLAB*. UAD Press.
- Masyhur.2004. *Penyelesaian Masalah Nilai Batas pada Persamaan Diferensial Biasa Orde Dua dengan Menggunakan Metode Beda Hingga (Skripsi)*. Yogyakarta: UAD Yogyakarta.
- Mathews, John dan Fink, Kurtis. 2004. *Numerical Methods Using MATLAB*. New Jersey: Pearson Education.
- Mohamad, M. Nor. 1993. *Pengenalan Persamaan Biasa*. Selangor: Percetakan Dewan Bahasa.
- Muhadjir, Noeng.1996. *Metodologi Penelitian Kualitatif*. Yogyakarta: Rake Sarasin.
- Soedijono, Bambang dan B. Susana.1993. *Model Matematik (Alat Penyusun Model)*. Jakarta: Karunia.

- Spiegel, Murray R.1983. *Matematika Lanjutan untuk Lanjutan Para Insinyur dan Ilmuwan* (Terj. Koko Martono). Jakarta: Erlangga.
- Sugiharto, Aris. 2006. *Pemrograman GUI dengan MATLAB*. Yogyakarta: Penerbit Andi.
- Triatmodjo, Bambang. 2002. *Metode Numerik*. Yogyakarta: Beta Offset.
- Utami, Murni.2005. *Penyelesaian Numeris Masalah Nilai Awal Menggunakan Metode Runge Kutta Berbasis MATLAB* (Skripsi).Yogyakarta: UIN Sunan Kalijaga Yogyakarta.
- Wardiman.1981. *Diktat Kuliah Persamaan Diferensial Elementer*. Yogyakarta: Jurusan FMIPA UGM Yogyakarta.

LISTING PROGRAM APLIKASI

```
function
% Edit the above text to modify the response to help
PROGRAM_APLIKASI
% Last Modified by GUIDE v2.5 07-Aug-2007 11:38:03
% Begin initialization code - DO NOT EDIT

handles = struct('gui_Name', 'filename', ...
                'gui_Singleton', 'gui_Singleton', ...
                'gui_OpeningFcn', '@PROGRAM_APLIKASI_OpeningFcn', ...
                ...
                'gui_OutputFcn', '@PROGRAM_APLIKASI_OutputFcn', ...
                ...
                'gui_LayoutFcn', [], ...
                'gui_Callback', []);

if nargin > 0
end

if nargin == 0
else
end

% End initialization code - DO NOT EDIT

% --- Executes just before PROGRAM_APLIKASI is made visible.
function

%function axes1_OpeningFcn(hObject, eventdata, handles, varargin)
%    % Choose default command line output for PROGRAM_APLIKASI
%
%    % Update handles structure
%
% --- Outputs from this function are returned to the command line.
function

% Get default command line output from handles structure

% --- Executes on button press in push_profil.
function
```

```
'profil.fig'

'Qomar.JPG'

% --- Executes on button press in push_ptnj.
function push_ptnj_Callback(hObject,eventdata,handles)
% hObject handle to push_ptnj; eventdata contains eventdata; handles
% structure of all handles to the figure

figure(handles.Figure,'Name','petunjuk.fig')
handles.petunjuk = hObject;
gui(handles.petunjuk);
handles.zim = guidata(handles.petunjuk,'zim.jpg');
handles.cursor = guidata(handles.petunjuk,'cursor');
axes(handles.axes1);
imshow(handles.zim);
hold on;
image(handles.cursor);

% --- Executes on button press in push_prog.
function push_prog_Callback(hObject,eventdata,handles)

figure(handles.Figure,'Name','METODE_BTM.fig')
handles.metode = hObject;
gui(handles.metode);

% --- Executes on button press in push_exit.
function push_exit_Callback(hObject,eventdata,handles)

text(handles.axes1,'Keluar Program')
Aplikasi?' 'Keluar' 'Ya' 'Tidak' 'Ya'
if strcmp(eventdata,'Ya')
    return
end
end
```

LISTING METODE BTM

```
function
% METODEDE_BTM M-file for METODEDE_BTM.fig
% Last Modified by GUIDE v2.5 08-Nov-2007 05:38:49
% Begin initialization code - DO NOT EDIT

    'gui_Name'           'METODEDE_BTM' ...
    'gui_Singleton'     '1' ...
    'gui_OpeningFcn'    'METODEDE_BTM_OpeningFcn' ...
    'gui_OutputFcn'     'METODEDE_BTM_OutputFcn' ...
    'gui_LayoutFcn'     'METODEDE_BTM_LayoutFcn' ...
    'gui_Callback'      []

if nargin > 2
    varargin{1} = hObject;
elseif nargin == 1
    hObject = gcf;
else
    hObject = [];
end

% End initialization code - DO NOT EDIT
% --- Executes just before METODEDE_BTM is made visible.
function

% Update handles structure

% --- Outputs from this function are returned to the command line.
function

% -----INPUT KOEFISIEN TEMBAKAN-----

function hObject = set_defaults(hObject, eventdata, handles)
    % hObject      handle to 'String' (see GCBO)
    % eventdata    reserved - to be defined in a future version of MATLAB
    % handles      structure with handles and user data (see GUIDATA)

    % Create handles for GUI components
    px.m
    set(hObject, 'px.m', 'w',
        'function y= f(x,y); \n'
        'y= %s; ');

% --- Executes during object creation, after setting all
properties.
function
if hObject ~= []
    set(hObject, 'BackgroundColor',
        'defaultUicontrolBackgroundColor')
end
```

```

                                'BackgroundColor' 'white'
end

function
    'String'
    x
                                'String'

    qx.m
    'qx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

% --- Executes during object creation, after setting all
properties.
function
if
                                'BackgroundColor'
    'defaultUicontrolBackgroundColor'
    'BackgroundColor' 'white'
end

function
    'String'
    x
                                'String'

    rx.m
    'rx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

% --- Executes during object creation, after setting all
properties.
function
if
                                'BackgroundColor'
    'defaultUicontrolBackgroundColor'
    'BackgroundColor' 'white'
end

% -----INPUT KOEFISIEN BEDA HINGGA-----
--

function
    'String'
    x
                                'String'

    px.m
    'pa.m' 'w'

```

```

        'function y= f(x,y); \n'
        'y= %s; '

% --- Executes during object creation, after setting all
properties.
function qm = qa.createFcnObject('function y= f(x,y); \n'
                                'y= %s; ', 'BackgroundColor'
                                'defaultUicontrolBackgroundColor'
                                'BackgroundColor' 'white'
                                'BackgroundColor' 'white'
end

function qa = qa.createFcnObject('function y= f(x,y); \n'
                                'y= %s; ', 'String'
                                'String'
                                'String'
                                'String'
                                'String'
                                'qa.m'
                                'qa.m' 'w'
                                'function y= f(x,y); \n'
                                'y= %s; '
                                'String'
                                'String'
end

% --- Executes during object creation, after setting all
properties.
function ra = ra.createFcnObject('function y= f(x,y); \n'
                                'y= %s; ', 'BackgroundColor'
                                'defaultUicontrolBackgroundColor'
                                'BackgroundColor' 'white'
                                'BackgroundColor' 'white'
end

function ra = ra.createFcnObject('function y= f(x,y); \n'
                                'y= %s; ', 'String'
                                'String'
                                'String'
                                'String'
                                'String'
                                'ra.m'
                                'ra.m' 'w'
                                'function y= f(x,y); \n'
                                'y= %s; '
                                'String'
                                'String'
end

% --- Executes during object creation, after setting all
properties.
function ra = ra.createFcnObject('function y= f(x,y); \n'
                                'y= %s; ', 'BackgroundColor'
                                'defaultUicontrolBackgroundColor'
                                'BackgroundColor' 'white'
                                'BackgroundColor' 'white'
end

% -----INPUT SYARAT BATAS-----

```

```

function
    'String'
    'String'
try
    'String'
    if
        %errorldg('Data bukan numeris','Salah');
    end
catch
    'Salah mengisikan data, data harus numeris
    !!!!' 'Salah'
end

% --- Executes during object creation, after setting all
properties.
function
    if
        'BackgroundColor'
        'defaultUicontrolBackgroundColor'
        'BackgroundColor' 'white'
    end

function
    'String'
    'String'
try
    'String'
    if
        %errorldg('Data bukan numeris','Salah');
    end
catch
    'Salah mengisikan data, data harus numeris
    !!!!' 'Salah'
end

% --- Executes during object creation, after setting all
properties.
function
    if
        'BackgroundColor'
        'defaultUicontrolBackgroundColor'
        'BackgroundColor' 'white'
    end

% -----INPUT NILAI BATAS-----

function
    'String'
    'String'
try
    'String'
    if
        'Data bukan numeris' 'Salah'
    end
end

```



```

catch
    'Salah mengisikan data, data harus numeris
!!!' 'Salah'
end
%if (a>=handles.b)
    % errordlg('                Nilai a < b ', 'Awaz !');
%end

% --- Executes during object creation, after setting all
properties.
function
    if
        'BackgroundColor'
        'defaultUicontrolBackgroundColor'
        'BackgroundColor' 'white'
    end

function
    'String'
        'String'
    try
        'String'
        if
            'Data bukan numeris' 'Salah'
        end
    catch
        'Salah mengisikan data, HARUS numeris !!!' 'Salah'
    end
    %if (b<=handles.a)
        % errordlg('                Nilai b > a ', 'Awaz !');
    %end

% --- Executes during object creation, after setting all
properties.
function
    if
        'BackgroundColor'
        'defaultUicontrolBackgroundColor'
        'BackgroundColor' 'white'
    end

% -----INPUT ITERASI-----

function
    'String'
        'String'
    try
        'String'
        if
            %errordlg('Data bukan numeris', 'Salah');
        end
    catch
        'Salah mengisikan data, data harus numeris
!!!' 'Salah'
    end
end

```

```

if
    'Nilai N harus lebih besar dari 0 !!!' 'Salah'
end

% --- Executes during object creation, after setting all
properties.
function
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if
    'BackgroundColor'
    'defaultUicontrolBackgroundColor'
    'BackgroundColor' 'white'
end

% --- Executes on button press in push_savei.
function
    'String'
    x
    'String'

    px.m
    'px.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

    'String'
    x
    'String'

    qx.m
    'qx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

    'String'
    x
    'String'

    rx.m
    'rx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

```

%-----Save i

PP QQ RR

```
    'String'
    x
    'String'

    pa.m
    'pa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    'get'
    'get'
    'dataObject handles'

    getObject 'String'
    x
    'pa=get(handles.edit1) 'String'
    handles.edit1
    'pa=edit'
    'set= qa.m'
    'handles.edit1 'qa.m' 'w'
    'function y= f(x,y); \n'
    'handles.edit1 'y= %s; '
    'get'
    'dataObject handles'

    getObject 'String'
    x
    'pa=get(handles.edit1) 'String'
    handles.edit1
    'pa=edit'
    'set= ra.m'
    'handles.edit1 'ra.m' 'w'
    'function y= f(x,y); \n'
    'handles.edit1 'y= %s; '
    'get'
    'dataObject handles'
```

%-----Save i YA

YB A B N

```
    'String'
    'ya=get(handles.edit1) 'String'
    handles.edit1
    'set= get'
    'handles.edit1 'String'
    'handles.edit1'
    'handles.edit1 'String'
    'String'
    'String'
    'String'
    'String'
```

```

%----- Save i
Handlesnya

%-----
handles.your
handles.your
handles.your
handles.your
handles.your
handles.your

%-----
handles.your = inputdlg('Nama file:' 'Simpan')
if isempty(handles.your)
    handles.your = handles.your;
else
    if strcmp(handles.your(1,1), '')
        'File dengan nama tersebut sudah
ada!' 'Peringatan' 'Timpa' 'Batal' 'Batal'
        if strcmp(handles.your(1,1), 'Batal')
            %Break
        else
            end
        end
    end

    fopen(handles.your(1,1), 'w')
    fprintf(handles.your(1,1), '%s \n', '1');
    fprintf(handles.your(1,1), '%s \n', '2');
    fprintf(handles.your(1,1), '%s \n', '3');
    fprintf(handles.your(1,1), '%s \n', '4');
    fprintf(handles.your(1,1), '%s \n', '5');
    fprintf(handles.your(1,1), '%s \n', '6');
    fprintf(handles.your(1,1), '%s \n', '7');
    fprintf(handles.your(1,1), '%s \n', '8');
    fprintf(handles.your(1,1), '%s \n', '9');
    fprintf(handles.your(1,1), '%s \n', '10');
    fprintf(handles.your(1,1), '%s \n', '11');
    fprintf(handles.your(1,1), '%s \n', '12');
    fprintf(handles.your(1,1), '%s \n', '13');
    fprintf(handles.your(1,1), '%s \n', '14');
    fprintf(handles.your(1,1), '%s \n', '15');
    fprintf(handles.your(1,1), '%s \n', '16');
    fprintf(handles.your(1,1), '%s \n', '17');
    fprintf(handles.your(1,1), '%s \n', '18');
    fprintf(handles.your(1,1), '%s \n', '19');
    fprintf(handles.your(1,1), '%s \n', '20');

end

% --- Executes on button press in push_call.
function push_call_Callback(hObject, eventdata, handles)
    handles.your = inputdlg('Nama file:' 'Buka')
    if isempty(handles.your)
        handles.your = handles.your;
    else
        if strcmp(handles.your(1,1), '')
            'File tidak ditemukan ...!' 'Error'
            %Break
        else
            end
        end
    end

    %-----
    %s' 'whitespace' '\n'

```

```

%-----Call PX QX
RX
    x      'String'
                'String'

    px.m
    edit('px.m' 'w'
        'function y= f(x,y); \n'
        'y= %s; ' edit)
    delete('px.m')
    guidata(hObject, handles);

set(hObject, 'String'
    edit('x'
        'handles.edit' 'String'
        handles.edit)
    delete('qx.m'
        edit('qx.m' 'w'
            'function y= f(x,y); \n'
            'y= %s; ' edit)
        delete('qx.m')
        guidata(hObject, handles);

set(hObject, 'String'
    edit('x'
        'handles.edit' 'String'
        handles.edit)
    delete('rx.m'
        edit('rx.m' 'w'
            'function y= f(x,y); \n'
            'y= %s; ' edit)
        delete('rx.m')
        guidata(hObject, handles);

%-----Call PP QQ
RR
set(hObject, 'String'
    edit('x'
        'handles.edit' 'String'
        handles.edit)
    edit('pa.m'
        edit('pa.m' 'w'
            'function y= f(x,y); \n'
            'y= %s; ' edit)
        delete('pa.m')
        guidata(hObject, handles);

    x      'String'

```

'String'

```
qa.m
    'qa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    handles=handles;
end

function 'String'
    x
    get(handles.edit_px 'String')
handles=handles;
relechar(ra)
delete ra.m
find=find('ra.m' 'w')
sprintf(find 'function y= f(x,y); \n')
sprintf(find 'y= %s; ')
close(find);
update(hObject,handles)

%-----Call
 hObject

    get(hObject 'String')
    hObject=find('edit_px' 'Tag' 'edit_px')
    set(hObject 'String' data{1,1})
    update(hObject,handles)
    handles=handles;

    get(hObject 'String')
    hObject=find('edit_qx' 'Tag' 'edit_qx')
    set(hObject 'String' data{1,1})
    update(hObject,handles)
    handles=handles;

    get(hObject 'String')
    hObject=find('edit_rx' 'Tag' 'edit_rx')
    set(hObject 'String' data{1,1})
    update(hObject,handles)
    handles=handles;

    get(hObject 'String')
    hObject=find('edit_pa' 'Tag' 'edit_pa')
    set(hObject 'String' data{1,1})
    update(hObject,handles)
    handles=handles;

    'String'
    'Tag' 'edit_qa'
    'String'
```

```

        'String'
        'Tag' 'edit_ra'
        'String'

        'String'
        'String'
        'Tag' 'edit_ya'
        'String'
        finddata(hObject,handles)
        handles.ya=ya;

        get(hObject,'String')
        yb=string(num2str(handles.yb))
        'String'
        hObject=find(handles,'Tag','edit_yb')
        set(hObject,'String',yb)
        guidata(hObject,handles)

        get(hObject,'String')
        handles.ya=ya;
        'String'
        hObject=find(handles,'Tag','edit_a')
        'String'
        finddata(hObject,handles)
        handles.a=a;

        get(hObject,'String')
        handles.yb=yb;
        'String'
        hObject=find(handles,'Tag','edit_b')
        'String'
        guidata(hObject,handles)
        handles.b=b;

        'String'
        handles.ya=ya;
        'String'
        hObject=find(handles,'Tag','edit_n')
        'String'
        guidata(hObject,handles)
        handles.n=n;
end

% --- Executes on button press in push_reset.
function push_reset(hObject, eventdata, handles)
handles.edit_ra('String',' ')
handles.edit_ya('String',' ')
handles.edit_a('String',' ')
handles.edit_b('String',' ')
handles.edit_n('String',' ')
handles.ya('String',' ')
handles.yb('String',' ')
handles.a('String',' ')
handles.b('String',' ')
handles.n('String',' ')

```

```

                                'String' ' '
    all
    px qx rx pa qa ra a b ya yb N

% --- Executes on button press in push_grafek.
function
%-----GRAFIK EKSAK

% --- Executes on button press in push_grafek.
% --- 'Grafik eksak sedang proses ...'
% --- 'Tunggu sebentar ... Jika GRAFIK EKSAK tidak
muncul,... lihat Command Window MATLAB. Jika di Command Window
MATLAB ada tulisan Error, (in dsolve/IC) The implicit option is
not available when giving Initial Conditions,berarti MNB tidak
bisa diselesaikan secara eksak (grafik juga tidak ada), coba
dengan grafik metode BedaHingga / Tembakan' 'Perhatian'

get(handles,'String')
x = x;
get(handles,'String')
handles.String = ' ';
delete(px.m)
fprintf('px.m' 'w'
'function y= f(x,y); \n'
'y= %s; '
handles.String);
delete(handles.String);

get(handles,'String')
x = x;
get(handles,'String')
handles.String = ' ';
delete(qx.m)
fprintf('qx.m' 'w'
'function y= f(x,y); \n'
'y= %s; '
handles.String);
delete(handles.String);

get(handles,'String')
x = x;
get(handles,'String')
handles.String = ' ';
delete(rx.m)
fprintf('rx.m' 'w'
'function y= f(x,y); \n'
'y= %s; '
handles.String);
delete(handles.String);

get(handles,'String')

```



```

    x
    'String'

    pa.m
    'pa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; ' %x
    %run(handles, handles)

    qa.m
    'qa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; ' %x
    %run(handles, handles)

    ra.m
    'ra.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; ' %x
    %run(handles, handles)

    %set(handles.edit1, 'String')
    handles.edit1
    %get(handles.edit1, 'String')
    handles.edit1
    %set(handles.edit2, 'String')
    handles.edit2
    %get(handles.edit2, 'String')
    handles.edit2
    %set(handles.edit3, 'String')
    handles.edit3
    %get(handles.edit3, 'String')
    handles.edit3

    % Run dari Call untuk ya, yb, a, b, dan N
    'String'

    'String'

    'String'

```

```

                                'String'
                                'String'

% handles.pa;
% handles.pa;
% handles.ra;

ya=handles.ya;
yb=handles.yb;
xa=handles.xa;
xb=handles.xb;

if (a>=handles.b)
    fprintf('          Nilai a < b ' 'Awas !'
end
%handles.x;

%handles
%handles

%get(handles.edit_rx 'String'
handles.rx=px;
%get(handles.edit_ox 'String'
handles.ox=ox;
%get(handles.edit_xr 'String'
handles.xr=rx;

ya=num2str(handles.edit_ya 'String'
handles.ya=ya;
yb=num2str(handles.edit_yb 'String'
handles.yb=yb;
xa=num2str(handles.edit_xa 'String'
handles.xa=xa;
xb=num2str(handles.edit_xb 'String'
handles.xb=xb;

%ans x
%lets eks.m
%fd=fopen('eks.m' 'w'
%fprintf('function vv = eks(x); \n'
%fprintf('    vv=dsolve(''D2y=%s*Dy%s*y%s,y(%s)=%s,y(%s)=%s'','x'
'); \n'
                                'String'
                                'String'

```

```

                                'eks'

                                x
                                eksku.m
                                % load file 'eksku.m' 'w'
                                % function Zeksku = eksku(x); \n'
                                % Zeksku=%s; \n' eksku.m'
                                % -----
                                %double(x)
                                %-----

                                for i=1:N
                                    eksku = eval('eksku')
                                end
                                %{
                                Exc=double(eksak)
                                xn=a:h:b;
                                xe=a+h:h:h*(N-1);
                                %}
                                %-----
                                %double(eksak)
                                %-----

                                fig = open('Grafik.fig')
                                handle = findobj('Parent','figure')
                                subplot(2,1,1)
                                plot(xn,Exc,'mp-') title('Grafik Penyelesaian Masalah Nilai Batas
                                '

                                hold on
                                lg = legend('eksak')
                                set(gcf,'Interpreter','none')
                                xlabel('Subinterval x(n)')
                                ylabel('Hasil Penyelesaian y[x(n)]')

                                %-----
                                %-----

                                % --- Executes on button press in push_grafbed.
                                function
                                %-----
                                -----GRAFIK BEDA HINGGA

                                    'Grafik beda hingga sedang proses ...'
                                    'Tunggu sebentar ... Jika GRAFIK BEDA HINGGA tidak
                                    muncul,... lihat Command Window MATLAB untuk diagnosa
                                    kesalahan' 'Perhatian'

```

```

    'String'
x
    'String'

    px.m
    'px.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    delete px.m
    rm -rf px.m
    'function y= f(x,y); \n'
    'y= %s; '
    delete px.m
    rm -rf px.m
    'function y= f(x,y); \n'
    'y= %s; '
    delete px.m
    rm -rf px.m

    qx.m
    'qx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    delete qx.m
    rm -rf qx.m
    'function y= f(x,y); \n'
    'y= %s; '
    delete qx.m
    rm -rf qx.m

    rx.m
    'rx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    delete rx.m
    rm -rf rx.m
    'function y= f(x,y); \n'
    'y= %s; '
    delete rx.m
    rm -rf rx.m

    pa.m
    'pa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    delete pa.m
    rm -rf pa.m
    'function y= f(x,y); \n'
    'y= %s; '
    delete pa.m
    rm -rf pa.m

    'String'
x
    'String'

    qa.m
    'qa.m' 'w'

```

```

        'function y= f(x,y); \n'
        'y= %s; '

    x = 'String'
    y = 'String'

    % Save file
    save('ra.m')
    % Open file
    fopen('ra.m' 'w')
    fprintf(fidl 'function y= f(x,y); \n'
        fprintf(fidl 'y= %s; ' y)
    fclose(fidl)
    guidata(hObject, handles)

    % Get handles
    handles.edit_x = 'String'
    handles.y_ax = 'String'
    handles.edit_y = 'String'
    handles.x_ax = 'String'
    handles.y_ax2 = 'String'

    handles.edit_xa = 'String'
    handles.y_ax2a = 'String'
    handles.edit_ya = 'String'
    handles.x_ax2 = 'String'
    handles.y_ax2a = 'String'

    % Run dari Call untuk ya, yb, a, b, dan N
    handles.ya = str2num(handles.edit_ya) 'String'
    handles.yb = str2num(handles.edit_yb) 'String'
    handles.a = str2num(handles.edit_xa) 'String'
    handles.a2 = str2num(handles.edit_xa2) 'String'
    handles.b = str2num(handles.edit_ya) 'String'
    handles.N = str2num(handles.edit_N) 'String'

    xl=handles.ya;
    xl=handles.yb;
    xl=handles.x;

```

```

if
    '                Nilai a < b ' 'Awat !'
end

%{
% -----FINITE DIFFERENCE METHOD ALTERNATIF 2
digits(7)

T = zeros(1,N+1);
y = zeros(1,N-1);
Va = zeros(1,N-2);
Vb = zeros(1,N-1);
Vc = zeros(1,N-2);
Vd = zeros(1,N-1);
% Sub-interval
h = (b-a)/N;
% Partisi Interval
%x1=a+h:h:a+h*(N-1);
x2=a+h:h:a+h*(N-1);

% Vektor konstan b dalam Ay=b
Vb(2:N-2) = vpa(h^2*feval('ra',x2(2:N-2)))
Vb(1) = vpa(h^2*feval('ra',x2(1))-(1+h/2*feval('pa',x2(1)))*ya)
Vb(N-1)= vpa(h^2*feval('ra',x2(N-1))-(1-h/2*feval('pa',x2(N-1)))*yb)

% Main diagonal dalam vektor A
Vd(1:N-1) = vpa(-(2+h^2*feval('qa',x2(1:N-1))))

% Calculate the superdiagonal of A
xa = x2(1,2:N-1);
Va(2:N-1) = vpa(1-h/2*feval('pa',xa))

% Calculate the subdiagonal of A
xc = x2(1,1:N-2);
Vc(1:N-2) = vpa(1-h/2*feval('pa',xc(1:N-2)))

% Solve using trisys
y= trisys(Va,Vd,Vc,Vb)
T = [a,x2,b]
y = [ya,y,yb]
F = [T' y']

%}

% -----FINITE DIFFERENCE METHOD ALTERNATIF 1

'pa'
'qa'

```

```

                                'ra'
%Subdiagonal pada matrik A
%Maindiagonal pada matrik A
%Superdiagonal pada matrik A
% bentuk matriknya

% penjumlahan matrik
% matrik b
% hasil penyelesaian MNB

% --- Executes on button press in push_graftemb.
function push_graftemb_Callback(hObject, eventdata, handles)
% hObject handle to push_graftemb (always 'self')
% eventdata structure containing the 'EventType' property
% handles structure of handles containing user data

figure_name = 'Grafik.fig';
figure_name = 'b^-'; title('Grafik Penyelesaian Masalah Nilai Batas');
hold on;
hold off;
title('beda_hingga');
set(gcf, 'Interpreter', 'none');
xlabel('Subinterval x(n)');
ylabel('Hasil Penyelesaian y[x(n)]');

% --- Executes on button press in push_graftemb.
function push_graftemb_Callback(hObject, eventdata, handles)
% hObject handle to push_graftemb (always 'self')
% eventdata structure containing the 'EventType' property
% handles structure of handles containing user data

title('Grafik tembakan sedang proses ...');
title('Tunggu sebentar ... Jika GRAFIK TEMBAKAN tidak muncul,... lihat Command Window MATLAB untuk diagnosa kesalahan');
title('Perhatian');

set(gcf, 'String', 'x');
set(gcf, 'String', 'String');

px.m
'px.m' 'w'

```

```

        'function y= f(x,y); \n'
        'y= %s; '

    'String'
    x

    'String'

    'qx.m'
    f=fopen('qx.m' 'w')
    fprintf(f,'function y= f(x,y); \n'
    fprintf(f,'y= %s; '
    fclose(f);
    guidata(hObject, handles);

    get(hObject, 'String')
    type x
    x=get(handles.edit1, 'String')
    handles.x=x;
    xi=xi+1;
    delete rx.m
    f=fopen('rx.m' 'w')
    fprintf(f,'function y= f(x,y); \n'
    fprintf(f,'y= %s; '
    fclose(f);
    guidata(hObject, handles);

    get(hObject, 'String')
    x
    pa=get(handles.edit1, 'String')
    handles.pa=pa;
    delete pa.m
    f=fopen('pa.m' 'w')
    fprintf(f,'function y= f(x,y); \n'
    fprintf(f,'y= %s; '
    fclose(f);
    guidata(hObject, handles);

    get(hObject, 'String')
    type x
    x=get(handles.edit1, 'String')
    handles.x=x;
    delete qa.m
    f=fopen('qa.m' 'w')
    fprintf(f,'function y= f(x,y); \n'
    fprintf(f,'y= %s; '

    'String'
    x

```



```

                                'String'

ra.m
    'ra.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; ' %a %b %N
    % Run dari Call untuk handles
    % Edit handles.edit_px 'String'
    handles.px=px;
    % Edit handles.edit_qx 'String'
    handles.qx=qx;
    % Edit handles.edit_rx 'String'
    handles.rx=rx;

    % Edit handles.edit_ypa 'String'
    handles.ypa=ypa;
    % Edit handles.edit_ypb 'String'
    handles.ypb=ypb;
    % Edit handles.edit_ypN 'String'
    handles.ypN=ypN;

% Run dari Call untuk ya, yb, a, b, dan N
y=call(handles,handles.edit_ypa, 'String'
handles.ya=ya;
y=call(handles,handles.edit_ypb, 'String'
handles.yb=yb;
a=call(handles,handles.edit_a, 'String'
handles.a=a;
b=call(handles,handles.edit_b, 'String'
handles.b=b;
N=call(handles,handles.edit_n, 'String'
handles.N=N;

rxl=handles.rx;
ryl=handles.ry;
ryr=handles.ry;
ryl=handles.ry;

% Edit handles.ya
handles.ya;
% Edit handles.yb
handles.yb;
handles.a;
handles.b;
handles.N;

if
    '
                                Nilai a < b ' 'Awas !'
end

```

```

%----- SHOOTING METHOD

    'String'

    F1.m
    fid=fopen('F1.m','w')
    fprintf(fid,'function Z= F1(x,Z); \n'
    fprintf(fid,'X=Z(1);Y=Z(2); \n'
    fprintf(fid,'Z=[ Y, %s*Y%s*X%s ]; \n',p1,p2,r1);
    fclose(fid);
delete F2.m
fid=fopen('F2.m','w')
fprintf(fid,'function Z= F2(x,Z); \n'
fprintf(fid,'X=Z(1);Y=Z(2); \n'
fprintf(fid,'Z=[ Y, %s*Y%s*X ]; \n',p1,p2);
fclose(fid);
format short;
clear;

% Menghitung F1
N=100;
h=1/N;
t=0;
Z=[1;0];
for k=1:N
    fprintf(fid,'F1' %s(1),Z(1),h);
    fprintf(fid,'F1' %s(1-h/2),Z(1),h);
    fprintf(fid,'F1' %s(1-h/2),Z(1),h);
    fprintf(fid,'F1' %s(1-h/2),Z(1),h);
    Z=[Z(1)+h*Z(2);k*Z(1)+k^2+Z(2)*k];
end
% save data
save('data1.mat','Z');

% Menghitung F2
N=100;
h=1/N;
t=0;
Z=[1;0];
for k=1:N
    fprintf(fid,'F2'
    fprintf(fid,'F2'
    fprintf(fid,'F2'
    fprintf(fid,'F2'

```

```
end
```

```
% HASIL MNB DENGAN METODE TEMBAKAN
```

```
figure('Name','Grafik.fig')
set(gcf,'Name','Grafik Penyelesaian Masalah Nilai Batas')
hold on;
plot(xm,'rs-');
title('Grafik Penyelesaian Masalah Nilai Batas');
axis([0 1 0 1]);
xlabel('Subinterval x(n)');
ylabel('Hasil Penyelesaian y[x(n)]');
set(gcf,'MenuBar','none');
set(gcf,'Name','tembakan');
set(gcf,'Interpreter','none');
set(gcf,'Label','Subinterval x(n)');
set(gcf,'YLabel','Hasil Penyelesaian y[x(n)]');
set(gcf,'ZLabel','');
set(gcf,'ZRotation',0);

% --- Executes on button press in push_allgrafik.
function
%-----GRAFIK ALL METHODS

% hObject      'Grafik all methods sedang proses ...'
% eventdata   'Sedang proses ... Jika grafik ALL METHODS tidak
muncul, lihat Command Window MATLAB. Jika di Command Window MATLAB
ada tulisan Error, (in dsolve/IC) The implicit option is not
available when giving Initial Conditions,berarti MNB tidak bisa
diselesaikan secara eksak, coba dengan grafik metode BedaHingga /
Tembakan' 'Perhatian'

% hObject      'String'
% eventdata   x
% eventdata   'String'
% eventdata   px.m
% eventdata   'px.m' 'w'
% eventdata   'function y= f(x,y); \n'
% eventdata   'y= %s; '
% eventdata   'String'
% eventdata   x
% eventdata   'String'
% eventdata   qx.m
```

```

    'qx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

    'String'
x
    'String'
    rx.m
    'rx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    'String'
    'String'
    'String'
    delete pa.m
    'pa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    'String'
    'handles'
    'String'
    x
    'String'
    'String'
    delete qa.m
    'qa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    'String'
    'handles'
    'String'
    x
    'String'
    'String'
    delete ra.m
    'ra.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

    'String'

```

```

    ya=string(get(handles,'a')); % 'String'
    yb=string(get(handles,'b')); % 'String'
    a=string(get(handles,'a')); % 'String'
    b=string(get(handles,'b')); % 'String'
    N=string(get(handles,'N')); % 'String'
    handles.ya=ya;
    handles.yb=yb;
    handles.a=a;
    handles.b=b;
    handles.N=N;

% Run dari Call untuk ya, yb, a, b, dan N
ya=string(get(handles,'a')); % 'String'
handles.ya=ya;
yb=string(get(handles,'b')); % 'String'
handles.yb=yb;
a=string(get(handles,'a')); % 'String'
handles.a=a;
b=string(get(handles,'b')); % 'String'
handles.b=b;
N=string(get(handles,'N')); % 'String'
handles.N=N;

p=handles.px;
q=handles.qx;
r=handles.rx;

P=handles.p;
q=handles.q;
r=handles.r;

ya=handles.y;
yb=handles.yb;
a=handles.a;
b=handles.b;

if (a<b) % 'Awasi !'
    error(' Nilai a < b ' 'Awasi !' )
end
%handles.N;

% click
%{
% -----FINITE DIFFERENCE METHOD ALTERNATIF 2
digits(7)

T = zeros(1,N+1);
y = zeros(1,N-1);
Va = zeros(1,N-2);
Vb = zeros(1,N-1);
Vc = zeros(1,N-2);
Vd = zeros(1,N-1);
% Sub-interval

```

```

h = (b-a)/N;
% Partisi Interval
%x1=a+h:h:a+h*(N-1);
x2=a+h:h:a+h*(N-1);

% Vektor konstan b dalam Ay=b
Vb(2:N-2) = vpa(h^2*feval('ra',x2(2:N-2)))
Vb(1) = vpa(h^2*feval('ra',x2(1))-(1+h/2*feval('pa',x2(1)))*ya)
Vb(N-1)= vpa(h^2*feval('ra',x2(N-1))-(1-h/2*feval('pa',x2(N-
1))))*yb)

% Main diagonal dalam vektor A
Vd(1:N-1) = vpa(-(2+h^2*feval('qa',x2(1:N-1))))

% Calculate the superdiagonal of A
xa = x2(1,2:N-1);
Va(2:N-1) = vpa(1-h/2*feval('pa',xa))

% Calculate the subdiagonal of A
xc = x2(1,1:N-2);
Vc(1:N-2) = vpa(1-h/2*feval('pa',xc(1:N-2)))

% Solve using trisys
y= trisys(Va,Vd,Vc,Vb)
T = [a,x2,b]
Y = [ya,y,yb]
F = [T' y']

%}

% -----FINITE DIFFERENCE METHOD ALTERNATIF 1

'pa'
'qa'
'ra'

%Subdiagonal pada matrik A

%Maindiagonalpada matrik A

%Superdiagonal pada matrik A

% bentuk matriknya

% penjumlahan matrik

% matrik b

%hasil penyelesaian MNB

```

```

%----- SHOOTING METHOD

%----- Menghitung F1

%----- Menghitung F2

for
    'F2'

```

```

                'F2'
                'F2'
                'F2'
end

%-----MNB DENGAN METODE TEMBAKAN
%-----MNB DENGAN METODE TEMBAKAN
%-----MNB DENGAN METODE TEMBAKAN
%-----MNB DENGAN METODE TEMBAKAN
%-----MNB DENGAN METODE TEMBAKAN
%-----MNB DENGAN METODE TEMBAKAN

%-----METODE EKSAK
xx=get(handles.edit_t_x) %String
handles.px=xx;
yy=get(handles.edit_t_y) %String
handles.py=yy;
zz=get(handles.edit_t_z) %String
handles.pz=zz;

yy=yy+1; %set handle t_y %String
handles.yy=yy;
yy=yy+1; %set handle t_y %String
handles.yy=yy;
a=handles.get(handles.edit_b) %String
handles.a=a;
b=handles.get(handles.edit_b) %String
handles.b=b;

%
x
delims = eks.m
fml = fopen('eks.m' 'w')
fprintf(fml, 'function vv = eks(x); \n'
        'vv=dsolve(''D2y=%s*Dy%s*y%s,y(%s)=%s,y(%s)=%s'', 'x'
        '); \n');
fclose(fml);
minidatadir('');

s=str2num(get(handles.edit_s) %String)
handles.s=s;
b=str2num(get(handles.edit_b) %String)
handles.b=b;

%
eks(x)
( b=a+1);
a=handles.get(handles.edit_s)
handles.s=a;
'eks'

x
eksku.m
'eksku.m' 'w'
'function Zeksku = eksku(x); \n'
'Zeksku=%s; \n'

```



```

for i=1:100
    [y,Exc]=solve('eksku', x2, N, h);
end
%{
Exc=double(eksak)
xn=a:h:b;
xe=a+h:h:h*(N-1);
%}
%--- Executes on button press in push_saveo.
figure('Name','Grafik.fig')
hold on
plot(xn, y, 'mp-', 'b^-', 'rs-', 'r')
title('Grafik Penyelesaian Masalah Nilai Batas ')
%--- Executes on button press in push_solve.
[eksak,beda_hingga,tembakan]=solve('eksak', 'beda_hingga', 'tembakan', 'Interpreter','none', 'Subinterval x(n)', 'Hasil Penyelesaian y[x(n)]')
%}
% --- Executes on button press in push_saveo.
function savefile('String', 'String')
% Save the current figure as a file.
% The first argument is the file name.
% The second argument is the file extension.

file_name='Nama file:' 'Simpan Hasil'
if isempty(file_name)
    file_name=input('Nama file: ');
end
if isempty(file_name)
    file_name='';
end
if ~isempty(file_name)
    file_name=[file_name '.fig'];
    if exist(file_name,'file')==2
        %File dengan nama tersebut sudah ada!
        'Peringatan' 'Timpa' 'Batal' 'Batal'
        if strcmp(file_name, 'Batal')
            %Break
        else
            end
        end
    end
    save(file_name, 'g');
end
% Save the current figure as a file.
% The first argument is the file name.
% The second argument is the file extension.
for i=1:100
    [y,Exc]=solve('eksku', x2, N, h);
end
% Save the current figure as a file.
% The first argument is the file name.
% The second argument is the file extension.
file_name=[file_name ' %s \n'];
end

```

```

end

end

% --- Executes on button press in push_print.
function
    'Nama file:' 'Cetak'
if
    if
        'File dengan nama tersebut sudah
ada!' 'Peringatan' 'Cetak' 'Batal' 'Batal'
        if
            'Batal'
                %Break
            else
                end
            end
        end
        'run.bat' 'wt'
        ' copy /b %s lpt1: \n' % Lx-800
(Dot metrik)
%     fprintf(fid,' copy /b %s USB001: \n',namafile); (USB)
%     fprintf(fid,' copy /b %s File: \n',namafile);
% end

!run
% print -f2 -dps 'Figure2.ps'
% print -dps 'Hasil2'
%namafile
%prindlg('-crossplatform',namafile);

end

run('-setup')

% --- Executes on button press in push_eksak.
function
%-----
-----RUN METODE EKSAK

        'Proses penyelesaian secara eksak ...'
        'Tunggu sebentar ... Jika penyelesaian EKSAK tidak
muncul di Figur METODE_BTM,... lihat Command Window MATLAB. Jika
di Command Window MATLAB ada tulisan Error, (in dsolve/IC) The
implicit option is not available when giving Initial
Conditions,berarti MNB tidak bisa diselesaikan secara eksak, coba
dengan metode BedaHingga / Tembakan' 'Perhatian'

        'String'
x
        'String'

```

```

px.m
    'px.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

get('px.m')
    'String'
x
get(handles.px.m) 'String'
handles.qx=qx;
del(handles.qx);
delete qx.m
fiddle=fopen('qx.m' 'w')
fprintf(fiddle 'function y= f(x,y); \n'
fprintf(fiddle 'y= %s; '
fclose(fiddle);
handles.qx=handles.qx;

get(handles.qx) 'String'
sym x
handles.qx=handles.qx; 'String'
handles.qx=qx;
delete qx.m
fiddle=fopen('rx.m' 'w')
fprintf(fiddle 'function y= f(x,y); \n'
fprintf(fiddle 'y= %s; '
fclose(fiddle);
handles.rx=handles.rx;

get(handles.rx) 'String'
sym x
handles.rx=handles.rx; 'String'
handles.rx=qx;
delete pa.m
fiddle=fopen('pa.m' 'w')
fprintf(fiddle 'function y= f(x,y); \n'
fprintf(fiddle 'y= %s; '
fclose(fiddle);
handles.pa=handles.pa;

get(handles.pa) 'String'
x
handles.pa=handles.pa; 'String'

qa.m
    'qa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

```

```

        'String'
    x
        'String'

    ra.m
        'ra.m' 'w'
        'function y= f(x,y); \n'
        'y= %s; '
        fprintf(
            fprintf(hObject, handles));
    px=get(handles.edit_px, 'String')
    handles.px=px;
    qx=get(handles.edit_qx, 'String')
    handles.qx=qx;
    rx=get(handles.edit_rx, 'String')
    handles.rx=rx;

    py=get(handles.edit_py, 'String')
    handles.py=py;
    qy=get(handles.edit_qy, 'String')
    handles.qy=qy;
    ry=get(handles.edit_ry, 'String')
    handles.ry=ry;

    % Run dari Call untuk ya, yb, a, b, dan N
    ya=str2num(get(handles.edit_ya, 'String'))
    handles.ya=ya;
    yb=str2num(get(handles.edit_yb, 'String'))
    handles.yb=yb;
    a=str2num(get(handles.edit_a, 'String'))
    handles.a=a;
    b=str2num(get(handles.edit_b, 'String'))
    handles.b=b;
    N=str2num(get(handles.edit_n, 'String'))
    handles.N=N;

    rxl=handles.rx;
    ryl=handles.py;
    rxl=handles.rx;
    ryl=handles.py;

    rxi=handles.px;
    ryi=handles.qy;
    rxi=handles.px;
    ryi=handles.qy;

    handles.rx;
    handles.qy;

    if
        '
        Nilai a < b ' 'Awat !'
    end

```

```

%-----METODE EKSAK
%-----
a=num2str(get(handles.edit_ya) 'String')
handles.ya=ya;
b=num2str(get(handles.edit_yb) 'String')
handles.yb=yb;
x=num2str(get(handles.edit_x) 'String')
handles.x=x;
a=num2str(get(handles.edit_a) 'String')
handles.a=a;
b=num2str(get(handles.edit_b) 'String')
handles.b=b;

save x
delete eks.m
fid=fopen('eks.m' 'w')
fprintf(fid 'function vv = eks(x); \n'
'vv=dsolve(''D2y=%s*Dy%s*y%s,y(%s)=%s,y(%s)=%s'',''x'
'); \n' x,yy,xa,ya,xb,yb)
fclose(fid);
delete(handles.figure);

a=num2str(get(handles.edit_a) 'String')
handles.a=a;
b=num2str(get(handles.edit_b) 'String')
handles.b=b;

vv=eks(x);

n=length(x);
x=x+0.01*(n-1);
eks1=vpaf('eks' x);
eksku=eks1';

save x
delete eksku.m

fid=fopen('eksku.m' 'w')
fprintf(fid 'function Zeksku = eksku(x); \n'
'Zeksku=%s; \n' eksku)

```

for


```

rx.m
    'rx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

get(handles.edit1) 'String'
x
    'handles.edit1pa' 'String'
    if (pa==pa)
        edit(handles.pa)
    delete pa.m
    fid=fopen('pa.m' 'w')
    fprintf(fid) 'function y= f(x,y); \n'
    fprintf(fid) 'y= %s; ' x
    fclose(fid);
    mkdir(handles.dirhandles);

set(handles.edit1) 'String'
sym x
ns=get(handles.edit1) 'String'
handles.dirhandles
mkdir(handles.dirhandles)
delete qa.m
fid=fopen('qa.m' 'w')
fprintf(fid) 'function y= f(x,y); \n'
fprintf(fid) 'y= %s; ' ns
fclose(fid);
mkdir(handles.dirhandles);

get(handles.edit2) 'String'
sym x
ns=get(handles.edit2) 'String'
handles.dirhandles
mkdir(handles.dirhandles)
delete ra.m
fid=fopen('ra.m' 'w')
fprintf(fid) 'function y= f(x,y); \n'
fprintf(fid) 'y= %s; ' ns
fclose(fid);
mkdir(handles.dirhandles);

x=get(handles.edit3) 'String'
handles.dirhandles
mkdir(handles.dirhandles)
x=get(handles.edit4) 'String'
handles.dirhandles
mkdir(handles.dirhandles)
x=get(handles.edit5) 'String'
handles.dirhandles
mkdir(handles.dirhandles)

handles.dirhandles
handles.dirhandles
handles.dirhandles

```



```

% Run dari Call untuk ya, yb, a, b, dan N
% ya = input('ya = ');
% yb = input('yb = ');
% a = input('a = ');
% b = input('b = ');
% N = input('N = ');

% handles edit_1
handles.edit_1 = edit('String');
% handles edit_2
handles.edit_2 = edit('String');
% handles edit_3
handles.edit_3 = edit('String');
% handles edit_4
handles.edit_4 = edit('String');
% handles edit_5
handles.edit_5 = edit('String');
% handles edit_6
handles.edit_6 = edit('String');
% handles edit_7
handles.edit_7 = edit('String');
% handles edit_8
handles.edit_8 = edit('String');
% handles edit_9
handles.edit_9 = edit('String');
% handles edit_10
handles.edit_10 = edit('String');
% handles edit_11
handles.edit_11 = edit('String');
% handles edit_12
handles.edit_12 = edit('String');
% handles edit_13
handles.edit_13 = edit('String');
% handles edit_14
handles.edit_14 = edit('String');
% handles edit_15
handles.edit_15 = edit('String');
% handles edit_16
handles.edit_16 = edit('String');
% handles edit_17
handles.edit_17 = edit('String');
% handles edit_18
handles.edit_18 = edit('String');
% handles edit_19
handles.edit_19 = edit('String');
% handles edit_20
handles.edit_20 = edit('String');
% handles edit_21
handles.edit_21 = edit('String');
% handles edit_22
handles.edit_22 = edit('String');
% handles edit_23
handles.edit_23 = edit('String');
% handles edit_24
handles.edit_24 = edit('String');
% handles edit_25
handles.edit_25 = edit('String');
% handles edit_26
handles.edit_26 = edit('String');
% handles edit_27
handles.edit_27 = edit('String');
% handles edit_28
handles.edit_28 = edit('String');
% handles edit_29
handles.edit_29 = edit('String');
% handles edit_30
handles.edit_30 = edit('String');
% handles edit_31
handles.edit_31 = edit('String');
% handles edit_32
handles.edit_32 = edit('String');
% handles edit_33
handles.edit_33 = edit('String');
% handles edit_34
handles.edit_34 = edit('String');
% handles edit_35
handles.edit_35 = edit('String');
% handles edit_36
handles.edit_36 = edit('String');
% handles edit_37
handles.edit_37 = edit('String');
% handles edit_38
handles.edit_38 = edit('String');
% handles edit_39
handles.edit_39 = edit('String');
% handles edit_40
handles.edit_40 = edit('String');
% handles edit_41
handles.edit_41 = edit('String');
% handles edit_42
handles.edit_42 = edit('String');
% handles edit_43
handles.edit_43 = edit('String');
% handles edit_44
handles.edit_44 = edit('String');
% handles edit_45
handles.edit_45 = edit('String');
% handles edit_46
handles.edit_46 = edit('String');
% handles edit_47
handles.edit_47 = edit('String');
% handles edit_48
handles.edit_48 = edit('String');
% handles edit_49
handles.edit_49 = edit('String');
% handles edit_50
handles.edit_50 = edit('String');
% handles edit_51
handles.edit_51 = edit('String');
% handles edit_52
handles.edit_52 = edit('String');
% handles edit_53
handles.edit_53 = edit('String');
% handles edit_54
handles.edit_54 = edit('String');
% handles edit_55
handles.edit_55 = edit('String');
% handles edit_56
handles.edit_56 = edit('String');
% handles edit_57
handles.edit_57 = edit('String');
% handles edit_58
handles.edit_58 = edit('String');
% handles edit_59
handles.edit_59 = edit('String');
% handles edit_60
handles.edit_60 = edit('String');
% handles edit_61
handles.edit_61 = edit('String');
% handles edit_62
handles.edit_62 = edit('String');
% handles edit_63
handles.edit_63 = edit('String');
% handles edit_64
handles.edit_64 = edit('String');
% handles edit_65
handles.edit_65 = edit('String');
% handles edit_66
handles.edit_66 = edit('String');
% handles edit_67
handles.edit_67 = edit('String');
% handles edit_68
handles.edit_68 = edit('String');
% handles edit_69
handles.edit_69 = edit('String');
% handles edit_70
handles.edit_70 = edit('String');
% handles edit_71
handles.edit_71 = edit('String');
% handles edit_72
handles.edit_72 = edit('String');
% handles edit_73
handles.edit_73 = edit('String');
% handles edit_74
handles.edit_74 = edit('String');
% handles edit_75
handles.edit_75 = edit('String');
% handles edit_76
handles.edit_76 = edit('String');
% handles edit_77
handles.edit_77 = edit('String');
% handles edit_78
handles.edit_78 = edit('String');
% handles edit_79
handles.edit_79 = edit('String');
% handles edit_80
handles.edit_80 = edit('String');
% handles edit_81
handles.edit_81 = edit('String');
% handles edit_82
handles.edit_82 = edit('String');
% handles edit_83
handles.edit_83 = edit('String');
% handles edit_84
handles.edit_84 = edit('String');
% handles edit_85
handles.edit_85 = edit('String');
% handles edit_86
handles.edit_86 = edit('String');
% handles edit_87
handles.edit_87 = edit('String');
% handles edit_88
handles.edit_88 = edit('String');
% handles edit_89
handles.edit_89 = edit('String');
% handles edit_90
handles.edit_90 = edit('String');
% handles edit_91
handles.edit_91 = edit('String');
% handles edit_92
handles.edit_92 = edit('String');
% handles edit_93
handles.edit_93 = edit('String');
% handles edit_94
handles.edit_94 = edit('String');
% handles edit_95
handles.edit_95 = edit('String');
% handles edit_96
handles.edit_96 = edit('String');
% handles edit_97
handles.edit_97 = edit('String');
% handles edit_98
handles.edit_98 = edit('String');
% handles edit_99
handles.edit_99 = edit('String');
% handles edit_100
handles.edit_100 = edit('String');

if (handles.a) < (handles.b)
    text('a < b', 'Nilai a < b ' 'Awat !')
end

%{
% -----FINITE DIFFERENCE METHOD ALTERNATIF 2
digits(7)

T = zeros(1,N+1);
y = zeros(1,N-1);
Va = zeros(1,N-2);
Vb = zeros(1,N-1);
Vc = zeros(1,N-2);
Vd = zeros(1,N-1);
% Sub-interval
h = (b-a)/N;
% Partisi Interval
%x1=a+h:h:a+h*(N-1);
x2=a+h:h:a+h*(N-1);

% Vektor konstan b dalam Ay=b
Vb(2:N-2) = vpa(h^2*feval('ra',x2(2:N-2)))
Vb(1) = vpa(h^2*feval('ra',x2(1))-(1+h/2*feval('pa',x2(1)))*ya)

```

```

Vb(N-1)= vpa(h^2*feval('ra',x2(N-1))-(1-h/2*feval('pa',x2(N-1)))*yb)

% Main diagonal dalam vektor A
Vd(1:N-1) = vpa(-(2+h^2*feval('qa',x2(1:N-1))))

% Calculate the superdiagonal of A
xa = x2(1,2:N-1);
Va(2:N-1) = vpa(1-h/2*feval('pa',xa))

% Calculate the subdiagonal of A
xc = x2(1,1:N-2);
Vc(1:N-2) = vpa(1-h/2*feval('pa',xc(1:N-2)))

% Solve using trisys
y= trisys(Va,Vd,Vc,Vb)
T = [a,x2,b]
Y = [ya,y,yb]
F = [T' y']

%}

% -----FINITE DIFFERENCE METHOD ALTERNATIF 1

                'Proses penyelesaian metode beda hingga...'
                'Tunggu sebentar ... Jika penyelesaian METODE BEDA
HINGGA tidak muncul di Figur METODE_BTM,... lihat Command Window
MATLAB untuk diagnosa kesalahan. ' 'Perhatian'

                'pa'
                'qa'
                'ra'

%Subdiagonal pada matrik A

%Maindiagonalpada matrik A

%Superdiagonal pada matrik A

% bentuk matriknya

% penjumlahan matrik

% matrik b

%hasil penyelesaian MNB

```



```

for
    %4.6f      |      '      |      %2.0f      |      %4.2f      |
    %4.6f      |      %4.6f      |      '      |
end

    % --- Executes on button press in push_temb.
function
%-----
-----RUN METODE TEMBAKAN
%-----Run dari Call untuk
koefisien

    'Proses penyelesaian metode tembakan...'
    'Tunggu sebentar ... Jika penyelesaian METODE BEDA
HINGGA tidak muncul di Figur METODE_BTM,... lihat Command Window
MATLAB untuk diagnosa kesalahan. ' 'Perhatian'

    'String'
    x
    'String'
    px.m
    'px.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    get(hObject,'String')
    x
    'String'
    qx.m
    'qx.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    'String'
    rx.m

```

```

    'rx.m' 'w'
    fidd=fopen('function y= f(x,y); \n'
    'y= %s; '
    'String'
x
    fidd=fopen('pa.m' 'String'
    'pa.m'
    fidd=fopen('pa.m' 'w'
    fprintf(fidd) 'function y= f(x,y); \n'
    fprintf(fidd) 'y= %s; '
    fclose(fidd);
    mkdir(handles,'pa.m');

get(handles,'String'
    yms x
    p=get(handles,'pa.m' 'String'
    handles,'pa.m');
    delete pa.m
    fidd=fopen('qa.m' 'w'
    fprintf(fidd) 'function y= f(x,y); \n'
    fprintf(fidd) 'y= %s; '
    fclose(fidd);
    mkdir(handles,'qa.m');

get(handles,'String'
    yms x
    r=get(handles,'ra.m' 'String'
    handles,'ra.m');
    delete ra.m
    fidd=fopen('ra.m' 'w'
    fprintf(fidd) 'function y= f(x,y); \n'
    fprintf(fidd) 'y= %s; '
    fclose(fidd);
    mkdir(handles,'ra.m');

xx=get(handles,'xx' 'String'
handles,xx=xx);
xx=get(handles,'xx' 'String'
handles,xx=xx);
xx=get(handles,'xx' 'String'
handles,xx=xx);

    'String'
    'String'
    'String'

```

```

% Run dari Call untuk ya, yb, a, b, dan N
get(handles.edit_1,'String')
get(handles.edit_2,'String')
get(handles.edit_3,'String')
get(handles.edit_4,'String')
get(handles.edit_5,'String')
get(handles.edit_6,'String')
get(handles.edit_7,'String')
get(handles.edit_8,'String')
get(handles.edit_9,'String')
get(handles.edit_10,'String')
get(handles.edit_11,'String')
get(handles.edit_12,'String')
get(handles.edit_13,'String')
get(handles.edit_14,'String')
get(handles.edit_15,'String')
get(handles.edit_16,'String')
get(handles.edit_17,'String')
get(handles.edit_18,'String')
get(handles.edit_19,'String')
get(handles.edit_20,'String')
get(handles.edit_21,'String')
get(handles.edit_22,'String')
get(handles.edit_23,'String')
get(handles.edit_24,'String')
get(handles.edit_25,'String')
get(handles.edit_26,'String')
get(handles.edit_27,'String')
get(handles.edit_28,'String')
get(handles.edit_29,'String')
get(handles.edit_30,'String')
get(handles.edit_31,'String')
get(handles.edit_32,'String')
get(handles.edit_33,'String')
get(handles.edit_34,'String')
get(handles.edit_35,'String')
get(handles.edit_36,'String')
get(handles.edit_37,'String')
get(handles.edit_38,'String')
get(handles.edit_39,'String')
get(handles.edit_40,'String')
get(handles.edit_41,'String')
get(handles.edit_42,'String')
get(handles.edit_43,'String')
get(handles.edit_44,'String')
get(handles.edit_45,'String')
get(handles.edit_46,'String')
get(handles.edit_47,'String')
get(handles.edit_48,'String')
get(handles.edit_49,'String')
get(handles.edit_50,'String')
get(handles.edit_51,'String')
get(handles.edit_52,'String')
get(handles.edit_53,'String')
get(handles.edit_54,'String')
get(handles.edit_55,'String')
get(handles.edit_56,'String')
get(handles.edit_57,'String')
get(handles.edit_58,'String')
get(handles.edit_59,'String')
get(handles.edit_60,'String')
get(handles.edit_61,'String')
get(handles.edit_62,'String')
get(handles.edit_63,'String')
get(handles.edit_64,'String')
get(handles.edit_65,'String')
get(handles.edit_66,'String')
get(handles.edit_67,'String')
get(handles.edit_68,'String')
get(handles.edit_69,'String')
get(handles.edit_70,'String')
get(handles.edit_71,'String')
get(handles.edit_72,'String')
get(handles.edit_73,'String')
get(handles.edit_74,'String')
get(handles.edit_75,'String')
get(handles.edit_76,'String')
get(handles.edit_77,'String')
get(handles.edit_78,'String')
get(handles.edit_79,'String')
get(handles.edit_80,'String')
get(handles.edit_81,'String')
get(handles.edit_82,'String')
get(handles.edit_83,'String')
get(handles.edit_84,'String')
get(handles.edit_85,'String')
get(handles.edit_86,'String')
get(handles.edit_87,'String')
get(handles.edit_88,'String')
get(handles.edit_89,'String')
get(handles.edit_90,'String')
get(handles.edit_91,'String')
get(handles.edit_92,'String')
get(handles.edit_93,'String')
get(handles.edit_94,'String')
get(handles.edit_95,'String')
get(handles.edit_96,'String')
get(handles.edit_97,'String')
get(handles.edit_98,'String')
get(handles.edit_99,'String')
get(handles.edit_100,'String')

if get(handles.edit_1) < get(handles.edit_2)
    fprintf('Nilai a < b ' 'Awes !')
end

%----- SHOOTING METHOD

get(handles.edit_101,'String')
px2=char(100);
py2=char(100);
rx2=char(100);
ry2=char(100);
delete F1.m
fid1=fopen('F1.m','w')
fprintf(fid1,'function Z= F1(x,Z); \n'
fprintf(fid1,'X=Z(1);Y=Z(2); \n'
fprintf(fid1,'Z=[ Y, %s*Y%s*X%s ]; \n',px2,py2,rx2,ry2);
fclose(fid1);
save F1.m
fid2=fopen('F2.m','w')
fprintf(fid2,'function Z= F2(x,Z); \n'
fprintf(fid2,'X=Z(1);Y=Z(2); \n'
fprintf(fid2,'Z=[ Y, %s*Y%s*X ]; \n',px2,py2);
fclose(fid2);

% Menghitung F1

```

```

for
    % F1
    k1=1; while k1<=n
        k2=1; while k2<=n-k1+1
            k3=1; while k3<=n-k1-k2+1
                vpa(2^k1+2^k2+2^k3-1)/6;
            end
            k3=k3+1;
        end
        k2=k2+1;
    end

% Menghitung F2
ab=1; while ab<=n
    b=1; while b<=n-ab+1
        c=1; while c<=n-ab-b+1
            2^c;
            c=c+1;
        end
        ab=ab+1;
    end

for k1=1:n
    k2=1; while k2<=n-k1+1
        k3=1; while k3<=n-k1-k2+1
            vpa(2^k1+2^k2+2^k3-1)/6;
        end
    end

% HASIL MNB DENGAN METODE TEMBAKAN
t1=vpa(1/2*(1+sqrt(5))-1);
t2=(t1+1);
tomb=double(t1);

ya=string(mn); handles.edit_text('String')
handles.ya=ya;
yb=string(mn); handles.edit_text('String')
handles.yb=yb;
zc=string(handles.edit_text('String'))
handles.zc=zc;
handles.edit_text('String')
handles.edit_text('String')

```

```

''
'
Penyelesaian Numeris Masalah Nilai Batas'
'
' Pada
Persamaan Diferensial Orde Dua Berbasis Komputasi'
' Studi
Komparatif antara Metode Beda Hingga dan Metode Tembakan'

'*****
*****
***'

'Persamaan : '
'
's*y %s' 'D2y = %s*Dy
'
'dengan'
' Koefisien p(x) = %s '
' Koefisien q(x) = %s '
' Koefisien r(x) = %s '
' Syarat batas : '
' y(a) = %5.2f '
' y(b) = %5.2f '
' Interval [a,b] : '
' a = %5.2f '
' b = %5.2f '
' Iterasi (N) = %5.2f '
' Size (h) = %5.5f '

'
'
% Tabel perbandingan eksak, beda hingga, tembakan
'
' Hasil penyelesaian metode tembakan adalah
: '
' -----
'
' | Iterasi | x(n) | y
' | waktu | |
' -----
'

for
'
' | %2.0f | %4.2f |
' | %4.6f | %4.6f |
'
end

'String'

% --- Executes on button press in push_allmethod.

```



```

function

%-----
-----RUN ALL METHODS
%-----Run dari Call untuk
koefisien

    'Tunggu sebentar ... Jika penyelesaian ALL METHODS
tidak muncul di Figur METODE_BTM,... lihat Command Window MATLAB.
Jika di Command Window MATLAB ada tulisan Error, (in dsolve/IC)
The implicit option is not available when giving Initial
Conditions,berarti MNB tidak bisa diselesaikan secara eksak, coba
dengan metode BedaHingga / Tembakan' 'Perhatian'

set(handles,'String')
axis x
xlabel(handles,'String')
handles_pxp=
plot(handles,
plot= px.m
title(handles,'px.m' 'w')
xlabel(handles,'function y= f(x,y); \n')
ylabel(handles,'y= %s; ')
hold on
subplot(handles,'String')

set(handles,'String')
axis x
xlabel(handles,'String')
handles_qx=
plot(handles,
plot= qx.m
title(handles,'qx.m' 'w')
xlabel(handles,'function y= f(x,y); \n')
ylabel(handles,'y= %s; ')
hold on
subplot(handles,'String')

set(handles,'String')
axis x
xlabel(handles,'String')
handles_rx=
plot(handles,
plot= rx.m
title(handles,'rx.m' 'w')
xlabel(handles,'function y= f(x,y); \n')
ylabel(handles,'y= %s; ')
hold on
subplot(handles,'String')

x
'String'
'String'

```

```

pa.m
    'pa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '

    'String'
x
    'pa.m' 'w'
    'function y= f(x,y); \n'
    'y= %s; '
    'String'
delete qa.m
fid=fopen('qa.m' 'w')
fprintf(fid, 'function y= f(x,y); \n')
fprintf(fid, 'y= %s; ' %a)
fclose(fid);
mkdir('home', 'handles');

set(hObject, 'String'
syms x
is=get(handles, 'String'
handles.pa.pa
handles.ra.ra
delete ra.m
fid=fopen('ra.m' 'w')
fprintf(fid, 'function y= f(x,y); \n')
fprintf(fid, 'y= %s; ' %a)
fclose(fid);
set(handles, 'handles');

px=get(handles, 'String'
handles.pa.pa
rx=get(handles, 'String'
handles.ra.ra
rx=set(handles, 'String'
handles.ra.ra

rx=get(handles, 'String'
handles.pa.pa
handles.ra.ra
rx=get(handles, 'String'
handles.ra.ra

% Run dari Call untuk ya, yb, a, b, dan N
ya=get(handles, 'String'
yb=get(handles, 'String'
a=get(handles, 'String'
b=get(handles, 'String'
N=get(handles, 'String'

```

```

% handles ya
% handles yb
% handles a1
% handles b1

if (a < handles.a1)
    error(' ')          Nilai a < b ' 'Awas !'
end

%{
% -----FINITE DIFFERENCE METHOD ALTERNATIF 2
digits(7)

T = zeros(1,N+1);
y = zeros(1,N-1);
Va = zeros(1,N-2);
Vb = zeros(1,N-1);
Vc = zeros(1,N-2);
Vd = zeros(1,N-1);
% Sub-interval
h = (b-a)/N;
% Partisi Interval
%x1=a+h:h:a+h*(N-1);
x2=a+h:h:a+h*(N-1);

% Vektor konstan b dalam Ay=b
Vb(2:N-2) = vpa(h^2*feval('ra',x2(2:N-2)))
Vb(1) = vpa(h^2*feval('ra',x2(1))-(1+h/2*feval('pa',x2(1)))*ya)
Vb(N-1)= vpa(h^2*feval('ra',x2(N-1))-(1-h/2*feval('pa',x2(N-1)))*yb)

% Main diagonal dalam vektor A
Vd(1:N-1) = vpa(-(2+h^2*feval('qa',x2(1:N-1))))

% Calculate the superdiagonal of A
xa = x2(1,2:N-1);
Va(2:N-1) = vpa(1-h/2*feval('pa',xa))

% Calculate the subdiagonal of A
xc = x2(1,1:N-2);
Vc(1:N-2) = vpa(1-h/2*feval('pa',xc(1:N-2)))

```

```

% Solve using trisys
y= trisys(Va,Vd,Vc,Vb)
T = [a,x2,b]
Y = [ya,y,yb]
F = [T' Y']

%}

% -----FINITE DIFFERENCE METHOD ALTERNATIF 1

% ----walaupun 'Proses penyelesaian metode beda hingga...'
%
% ----Membaca file
% ----membaca file
% ----membaca file 'pa'
% ----membaca file 'qa'
% ----membaca file 'ra'

%Subdiagonal pada matrik A

%Maindiagonalpada matrik A

%Superdiagonal pada matrik A

% bentuk matriknya

% ----Membaca file

% penjumlahan matrik

% matrik b

% ----Membaca file
% ----membaca file 'pa'
% ----membaca file 'qa'
% ----membaca file 'ra'

%hasil penyelesaian MNB
%
% ----Membaca file
% ----membaca file
% ----membaca file

%----- SHOOTING METHOD

% ----Membaca file 'String'
% ----membaca file
% ----membaca file
% ----membaca file

F1.m
'F1.m' 'w'
'function Z= F1(x,Z); \n'
'X=Z(1);Y=Z(2); \n'
'Z=[ Y, %s*Y%s*X%s ]; \n'

F2.m
'F2.m' 'w'
'function Z= F2(x,Z); \n'

```

```

        'X=Z(1);Y=Z(2); \n'
        'Z=[ Y, %s*Y%s*X ]; \n'

% Menghitung F1
N=1;
while (N<=Nmax)
    Z=[ Z(1),N+1];
    Z=[ Z(2),N+1,length(Z(2))];
    Z=[ Z(3),N+1];
    Z=[ Z(4),N+1,length(Z(4))];
    for i=1:4
        k1=vpaint('F1' 'Tembakan 1, Z1, Z2, Z3, Z4');
        k2=vpaint('F1' 'Tembakan 2, Z1, Z2, Z3, Z4');
        k3=vpaint('F1' 'Tembakan 3, Z1, Z2, Z3, Z4');
        k4=vpaint('F1' 'Tembakan 4, Z1, Z2, Z3, Z4');
        Z1=[Z(1),N+1];Z2=[Z(2),N+1];Z3=[Z(3),N+1];Z4=[Z(4),N+1];
    end
    v=vpaint('F1' 'Tembakan 5, Z1, Z2, Z3, Z4');
    closeall;

% Menghitung F2
N=1;
while (N<=Nmax)
    Z=[ Z(1),N+1];
    Z=[ Z(2),N+1,length(Z(2))];
    Z=[ Z(3),N+1];
    Z=[ Z(4),N+1,length(Z(4))];
    for i=1:4
        k1=vpaint('F2' 'Tembakan 1, Z1, Z2, Z3, Z4');
        k2=vpaint('F2' 'Tembakan 2, Z1, Z2, Z3, Z4');
        k3=vpaint('F2' 'Tembakan 3, Z1, Z2, Z3, Z4');
        k4=vpaint('F2' 'Tembakan 4, Z1, Z2, Z3, Z4');
        Z1=[Z(1),N+1];Z2=[Z(2),N+1];Z3=[Z(3),N+1];Z4=[Z(4),N+1];
    end
    v=vpaint('F2' 'Tembakan 5, Z1, Z2, Z3, Z4');
    v=exp(Y*(N+1));
% HASIL MNB DENGAN METODE TEMBAKAN
    v=exp(Y*(N+1));
end
    v=exp(Y*(N+1));

'Proses penyelesaian secara analitis ...'

```

```

                                'String'

                                'String'

                                'String'

                                handles.edit_yo 'String'

                                handles.edit_yr 'String'
                                if yf=yo;
                                handles.edit_a 'String'
                                handles.a=a;
                                handles.edit_ib 'String'
                                handles.ib;

%-----METODE EKSAK

syms x
delete eks.m
fid=fopen('eks.m' 'w')
fprintf(fid 'function vv = eks(x); \n'
'vv=dsolve(''D2y=%s*Dy%s*y%s,y(%s)=%s,y(%s)=%s'', 'x'
'); \n'
'end');
fclose(fid);
handles=guidObject, handles;

handles.edit_yo 'String'
handles.a=a;
handles.edit_ib 'String'
handles.ib;

vv=eval('eks');
h = figure('Name','EKSAK');
axes(handles.gca);
eksa=plot(vv, 'eks')
eksa=axis('off')

syms x
delete eksku.m
fid=fopen('eksku.m' 'w')
fprintf(fid 'function Zeksku = eksku(x); \n'
'Zeksku=%s; \n'
'end');
fclose(fid);
handles=guidObject, handles;

end
end
for
                                'eksku'

end

```

```

a=get(handles.edit_1,'String')
b=get(handles.edit_2,'String')
c=get(handles.edit_3,'String')
d=get(handles.edit_4,'String')
r=get(handles.edit_5,'String')
N=get(handles.edit_6,'String')
h=get(handles.edit_7,'String')
x0=a,x2=b;

eksk=[0];
eksk(1)=c;

Penyelesaian Numeris Masalah Nilai Batas'

Persamaan Diferensial Orde Dua Berbasis Komputasi'

Komparatif antara Metode Beda Hingga dan Metode Tembakan'

*****
*****
***!
'Persamaan : '
' D2y = %s*Dy
%s*y %s'
'dengan'
' Koefisien p(x) = %s '
' Koefisien q(x) = %s '
' Koefisien r(x) = %s '
' Syarat batas : '
' y(a) = %5.2f '
' y(b) = %5.2f '
' Interval [a,b] : '
' a = %5.2f '
' b = %5.2f '
' Iterasi (N) = %5.2f '
' Size (h) = %5.5f '

% Tabel perbandingan eksak, beda hingga, tembakan
' a. Tabel nilai eksak, beda hingga dan
tembakan : '
-----
-----
-----

```

```

eksak          |      | Iterasi |      x(n)      |      y
waktu         |      | y beda hingga |      y tembakan |
-----|-----|-----|-----|-----
-----|-----|-----|-----|-----
'Proses penyelesaian secara analitis ...'
for k=1:N
    t=timetock(t0);
    [Yk]=sprand(1,1,N);
    [Yk]=sprand(1,1,N);
    %4.6f      |      | %2.0f      |      %4.2f      |
    %4.6f      |      | %4.6f      |      %4.6f      |
end
% Tabel perbandingan error beda hingga dan tembakan
% b. Tabel nilai error mutlak dan error
relatif terhadap nilai eksak :
-----|-----|-----|-----|-----
'Error relatif' |      | Error mutlak |      |
'beda hingga'  |      | tembakan     |      |
'beda hingga'  |      | waktu        |      |
-----|-----|-----|-----|-----
'
close(t0);
end

% 'Proses penyelesaian secara analitis ...'
for k=1:N
    t=timetock(t0);
    [Yk]=sprand(1,1,N);
    [Yk]=sprand(1,1,N);
    %4.6f      |      | %2.0f      |      %4.2f      |
    %4.6f      |      | %4.6f      |      %4.6f      |
end

```



```

                                'String'
    %}

    % --- Executes on button press in push_help.
    function
                                'Help.fig'

    % --- Executes on button press in push_exit.
    function
                                'Keluar Program BHT?' 'Keluar' 'Ya' 'Tidak' 'Ya'
    if
                                'Tidak'
        return
    end

    % --- Executes on selection change in listbox1.
    function

    % --- Executes during object creation, after setting all
    properties.
    function
    if
                                'BackgroundColor'
                                'defaultUicontrolBackgroundColor'
                                'BackgroundColor' 'white'
    end

end

```

CURRICULUM VITAE

A. Data Diri

- a. Nama Lengkap : M. Kamari
- b. Jenis Kelamin : Laki-laki
- c. Tempat/Tgl. Lahir : Jepara, 8 Mei 1984
- d. Kewarganegaraan : Indonesia
- e. Agama : Islam
- f. Anak ke : Dua
- g. Status dalam keluarga : Anak Kandung
- h. Golongan darah : B
- i. Alamat asal : Bawu Ngemplak RT 18/04 Batealit Jepara
- j. Alamat Yogyakarta : Jl. Nyi Pembayun 21 Karang Prenggan Kotegede
- k. No.HP : 085292076084

B. Data Keluarga

- a. Nama Orang Tua
 - Ayah : Sarito (alm)
 - Ibu : Suratmi
- b. Pekerjaan Orang Tua
 - Ayah : -
 - Ibu : Ibu rumah tangga

C. Riwayat Pendidikan

- a. SDN Bawu IV Batealit Jepara : Lulus tahun 1997
- b. MTsN Bawu Batealit Jepara : Lulus tahun 2000
- c. MAN Bawu Jepara : Lulus tahun 2003
- d. S1 Pend. Matematika UIN Sunan Kalijaga Yogyakarta : Lulus tahun 2008