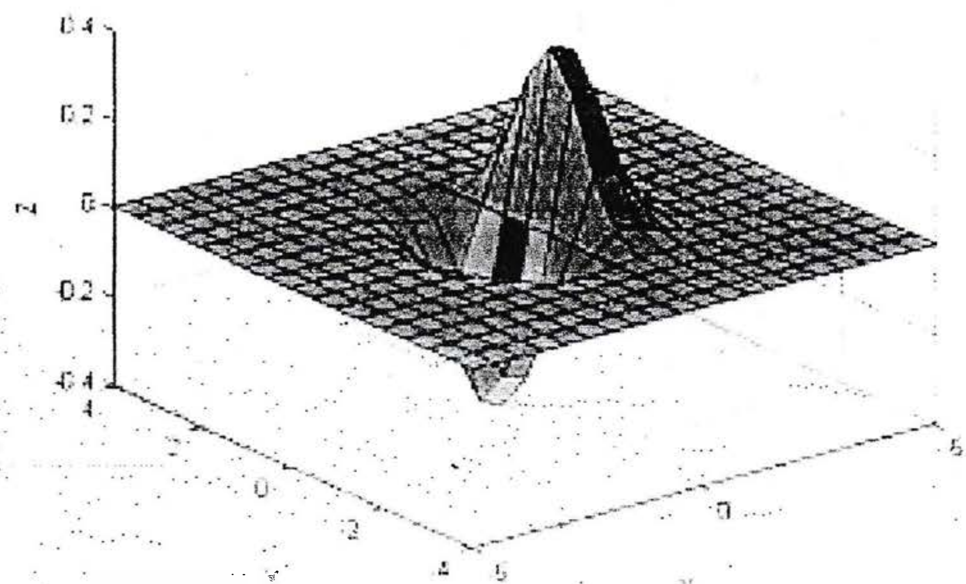
MODUL PRAKTIKUM

METODE NUMERIK



Pelaksana Praktikum:

Moranain Mungkin, ST, M.Si

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MEDAN AREA (UMA)
2018

DAFTAR ISI

DAFTAR ISI i

Tujuan Praktikum 1
Manfaat Praktikum 1
Teori Dasar Matlab 2
1 Matlab 2
1.1 Bagian dari Sistem Matlab 3
2. Matrik dalam Matlab 4
2.1. Membuat Matriks 4
2.2. Operasi Komputasi Matriks 6
2.3 Fungsi Matriks pada Matlab 7
3. Memulai Script Matlab 8
4 Aljabar Matriks 9
4.1 Penjumlahan dan Pengurangan Matriks 9
4.2 Perkalian Matriks 9
4.3. Matriks Identitas 9
4.4 Determinan 10
4.5. Invers 10
5. Persamaan Linier 10
5.1. Metode Eliminasi Gauss 10
5.2. Metode Eliminasi Gauss-Jordan 11
5.3 Metode Dekomposisi L-U 12
6. Praktik Mandiri 20

A. Tujuan Pratikum

1. Mahasiswa mampu mengoperasikan MATLAB dan memanfaatkannya sebagai perangkat Simulasi untuk praktikum Sinyal dan Sistem
2. Mahasiswa dapat menyelesaikan pemasalahan dalam metode numerik dengan menggunakan sistem MATLAB
3. Mempelajari penggunaan sistem help untuk mengetahui commands dan syntax dasar MATLAB
4. Dapat menggunakan MATLAB untuk desain filter

B. Manfaat Praktikum

Adapun maksud dan tujuan dari pembuatan laporan ini adalah :

- Mampu melakukan perhitungan matematis dengan menggunakan matlab.
- Mampu menggambarkan plot/grafik dari suatu fungsi atau persamaan menggunakan matlab
- Mampu menyelesaikan persamaan *linear* seperti metode metode Dekomposisi LU, metode Gauss dan metode Gauss-Jordan.
- Mampu menyelesaikan perhitungan bangun ruang secara cepat.

TEORI DASAR

1. MATLAB

MATLAB kependekan dari *MATrix LABoratory* merupakan sebuah paket perangkat lunak untuk komputasi teknik dan scientific (operasi-operasi matriks dan matematika, baik dalam aljabar maupun bilangan kompleks, fungsi-fungsi matriks, analisis data, polinomial, pengintegralan, pendeferensialan, persamaan-persamaan nonlinear, interpolasi, pemrosesan sinyal, dll). MATLAB juga telah memiliki sejumlah perintah yang siap pakai (Built-in), baik berupa variabel, pernyataan, maupun fungsi yang dapat langsung digunakan.

MATLAB bisa sebagai kalkulator dan bahasa pemrograman. Operasi yang dilakukan MATLAB adalah skalar, matriks dan vektor, serta teks.

MATLAB adalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang-bidang:

- Matematika dan Komputasi
- Pembentukan Algorithm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototipe
- Analisa data, explorasi, dan visualisasi
- Grafik Keilmuan dan bidang Rekayasa.

MATLAB merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu array sehingga tidak lagi kita dipusingkan dengan masalah dimensi. Hal ini memungkinkan kita untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, kususnya yang berhubungan dengan matrix dan formulasi vektor, yang mana masalah tersebut merupakan momok apabila kita harus menyelesaikannya dengan menggunakan bahasa level rendah seperti Pascall, C dan Basic. Nama MATLAB merupakan singkatan dari matrix laboratory. MATLAB pada awalnya ditulis untuk memudahkan akses perangkat lunak matrik yang telah dibentuk oleh LINPACK dan EISPACK. Saat ini perangkat MATLAB telah menggabung dengan LAPACK dan BLAS library, yang merupakan satu kesatuan dari sebuah seni tersendiri dalam perangkat lunak untuk komputasi matrix. Dalam lingkungan perguruan tinggi teknik, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Di industri, MATLAB merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tingi, pengembangan dan analisanya. Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih kita kenal dengan nama toolbox. Sangat penting bagi seorang pengguna Matlab, toolbox mana yang mandukung untuk learn dan apply technologi yang sedang dipelajarinya. Toolbox toolbox ini merupakan kumpulan dari fungsi-fungsi MATLAB (M-files) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas particular. Area-area yang sudah bisa dipecahkan dengan toolbox saat ini meliputi pengolahan sinyal, system kontrol, neural networks, fuzzy logic, wavelets, dan lain-lain.

1.1. Bagian dari Sistem MATLAB

Sebagai sebuah sistem, MATLAB tersusun dari 5 bagian utama:

1. Development Environment. Merupakan sekumpulan perangkat dan fasilitas yang membantuanda untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah graphical user interfaces (GUI). Termasuk didalamnya adalah MATLAB desktop dan Command Window, command history, sebuah editor dan debugger, dan browsers untuk melihat help, workspace, files, dan search path.
2. MATLAB Mathematical Function Library. Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar sepertri: sum, sin, cos, dan complex arithmetic, sampai dengan fungsi-fungsi yang lebih kompek seperti matrix inverse, matrix eigenvalues, Bessel functions, dan fast Fourier transforms.
3. MATLAB Language. Merupakan suatu high-level matrix/array language dengan control flow statements, functions, data structures, input/output, dan fitur-fitur object-oriented programming. Ini memungkinkan bagi kita untuk melakukan kedua hal baik "pemrograman dalam lingkup sederhana " untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang komplek.
4. Graphics. MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan high-level functions (fungsi-fungsi level tinggi) untuk visualisasi data dua dikensi dan data tiga dimensi, image processing, animation, dan presentation graphics. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi anda untuk membiasakan diri untuk memunculkan grafik mulai dari benutk yang sederhana sampai dengan tingkatan graphical user interfaces pada aplikasi MATLAB anda.
5. MATLAB Application Program Interface (API). Merupakan suatu library yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinterakasi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan routines dari MATLAB (dynamic linking), pemanggilan MATLAB sebagai sebuah computational engine, dan untuk membaca dan menuliskan MAT-files.

2. Matriks dalam MATLAB

2.1. Membuat Matriks

1. Per elemen

- menggunakan **spasi** untuk **memisahkan elemen** dalam suatu baris
- menggunakan tanda **semicolon (;)** untuk **memisahkan baris dengan baris berikutnya**
- elemen-elemen matrik diletakkan di antara tanda [dan]
- Contoh:
  >> A = **[1 2 3; 4 5 6; 7 8 9]**
- lalu tekan ENTER

```
>> A=[1 2 3; 4 5 6; 7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9
```

Untuk matrik dengan ukuran besar dapat dinyatakan ke dalam beberapa baris input dengan *carriage return* (ENTER) sebagai pengganti tanda semikolon(;)

2. Menggunakan fungsi FOR dan WHILE

Contoh:

```
>> For i = 1:3,
      For j = 1:3,
            a(i,j) = 4*i-(3+j);
      end
   end
```

Program tersebut berarti didefinisikan i dari 1 sampai 3 yang merupakan baris dari matrik dan kemudian juga didefinisikan j dari 1 sampai 3 yang merupakan kolom matrik. Kemudian dibuat matrik a yang setiap elemen-nya merupakan hasil penambahan dari i dan j sesuai looping yag berjalan.

Untuk mengetahui hasilnya maka diketikkan a yang merupakan variable penampung hasil eksekusi program :

>> *a*

```
>> for i = 1:3,
for j = 1:3,
a(i,j) = 4*(i-1)+j-1;
end
end
>> a

a =

     0     1     2
     4     5     6
     8     9    10
```

3. Menggunakan rutin yang ada dalam MATLAB

a. Matriks identitas ordo nxn

\>> **eye (n)**

```
>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1
```

b. Matriks satuan ordo nxn (semua elemennya bernilai 1 (satu))

\>> **ones(n)**

```
>> ones(3)

ans =

     1     1     1
     1     1     1
     1     1     1
```

c. Matrik dengan elemen acak ordo nxn (nilai elemen antara 0 dan 1)

\>> **rand (n)**

```
>> rand(3)

ans =

    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
```

d. Matrik dengan elemen berupa bilangan segitiga Pascal ordo nxn

\>> **pascal (n)**

```
>> pascal(3)

ans =

     1     1     1
     1     2     3
     1     3     6
```

e. Rutin yang lain:

Rutin	Keterangan
[ ]	matriks kosong
Company	matriks gabungan
Gallery	beberapa matriks pengujian yang kecil
Hadamard	matriks Hadamard
Hankel	matriks Hankel
Hilb	matriks Hilbert
Invhilb	invers matriks Hilbert
Magic	magic square
Randn	matriks random terdistribusi normal dengan elemen-elemennya memiliki mean nol dan varians satu
Rosser	matriks pengujian nilai eigen simetrik
Toeplitz	matriks toeplitz
Vander	matriks vandermonde
Wilkinson	matriks pengujuan nilai eigen Wilkinson...
Zeros	Matriks yang semua elemennya nol

2.2. Operasi Komputasi Matriks

Simbol	Operasi
*	Perkalian
/	Pembagian kanan (matriks)
\	Pembagian kiri (matriks)
^	Pemangkatan
+	Penjumlahan
-	Pengurangan
‘	Untuk mencari transpose matriks

Sama seperti komputasi manual, komputasi pada MATLAB memiliki prioritas dengan urutan perkalian atau pembagian, baru diikuti penjumlahan dan pengurangan. Jika ingin memprioritaskan operasi tertentu, bisa dilakukan dengan memberikan tanda kurung “( )”.

2.3. Fungsi Matriks pada MATLAB

Fungsi	Keterangan
`balance(A)`	Penyekalaan untuk memperbaiki akurasi nilai eigen
`cdf2rdf(A)`	bentuk diagonal kompleks ke bentuk diagonal blok real
`chol(A)`	faktorisasi Cholesky
`cond(A)`	matriks bilangan kondisi
`condest(A)`	estimasi matriks bilangan kondisi I-norm
`d=eig(A)` `[V,D]=eig(A)`	nilai eigen dan vektor eigen
`det(A)`	determinan
`expm(A)`	matriks eksponensial
`expm1(A)`	implementasi M-file dari expm
`exmp2(A)`	Matriks eksponensial menggunakan deret Taylor
`exmp3(A)`	Matriks eksponensial menggunakan nilai eigen dan eigen
`funm(A, 'fun')`	menghitung fungsi matriks umum
`hess(A)`	bentuk hessenberg
`invs(A)`	invers matriks
`logm(A)`	logaritma matriks
`lscov(A, b, V)`	kuadrat terkecil dengan kovarians yang diketahui
`lu(A)`	faktor dari eliminasi Gaussian
`nnls(A, b)`	kuadrat terkecil nonnegative
`norm(A)`	norm matriks dan vektor
`norm(A, 1)`	I-norm
`norm(A, 2)`	2-norm (Euclidean)
`norm(A, inf)`	takberhingga (infinity)
`norm(A, p)`	P-norm(hanya untuk vektor)
`norm(A, 'fro')`	F-norm
`null(A)`	spasi kosong
`orth(A)`	Ortogonalisasi
`pinv(A)`	Pseudoinvers
`poly(A)`	Variabeln karakteristik / mencari koefisien persamaan polinomial
`roots(A)`	mencari akar persamaan polinomial
`polyvalm(A)`	evaluasi polinomial matriks
`qr(A)`	dekomposisi ortogonal-triangular
`qrdelete(Q,R,j)`	menghapus kolom dari faktorisasi qr
`qrinsert(Q,R,j,x)`	menyelipkan kolom pada faktorisasi qr
`qz(A)`	nilai eigen yang digeneralisasi
`rank(A)`	banyaknya baris atau kolom yang independen linier
`recond(A)`	estimator kondisi resiprokal
`rref(A)`	mengurangi baris bentuk echelon
`rsf2csf`	bentuk schur real ke bentuk schur kompleks
`schur(A)`	dekomposisi Schur

Fungsi	Keterangan
`sqrtm(A)`	matriks akar kuadrat
`svd(A)`	dekomposisi *nilai singular*
`trace(A)`	jumlah elemen diagonal

3. Memulai Script MATLAB

MATLAB menyediakan fasilitas makro, yang disebut M-file MATLAB karena *ekstension filenya* M. Dengan *fasilitas makro ini pemrograman* terhadap rutin-rutinnya dapat dilakukan sendiri oleh pemakai. Script file merupakan file yang berisi sekumpulan instruksi. Jika file ini dijalankan, maka instruksi-instruksi tersebut akan dijalankan secara berurutan. Dengan menuliskan nama file, kita dapat memanggil isi file tersebut.

a. **Dengan EDITOR DOS**
- tuliskan !edit <ENTER>
- tuliskan isi file
- simpanlah file
- keluar dari EDITOR DOS
- untuk memanggil, ketik *nama file* lalu tekan Enter.

b. **Dengan NOTEPAD**
- dengan menggunakan mouse, klik di ***File*** ➔ ***New*** ➔ ***M-file***
- tuliskan isi file
- simpanlah file pada direktori BIN dengan tahapan-tahapan berikut
  - untuk pilihan FILE NAME, isilah dengan nama dari script-file beserta ekstension-nya. Adapun ekstension dari script-file Matlab adalah .M , contoh : data . m
  - untuk pilihan SAVE AS TYPE, pilihlah : ALL FILES (*.*).
  - lalu klik-lah pilihan SAVE.
- keluar dari NOTEPAD
- untuk memanggil klik di File ➔ Run ➔ M-file, ketik nama file lalu klick OK, atau dapat juga dengan langsung mengetikkan nama dari Script-filenya.

Hal yang Harus Diperhatikan:

1. MATLAB hanya dapat digunakan untuk matrik-matrik persegi panjang dengan elemen bilangan kompleks.
2. Bila bagian imaginer bernilai nol maka tidak akan dicetak tetapi masih disediakan tempat di memori.
3. Matrik 1x1 dianggap sebagai skalar.
4. Matrik 1xn dianggap vektor baris.
5. Matrik mx1 dianggap vektor kolom
6. MATLAB adalah software yang ***case sensitive***, jadi huruf besar dan huruf kecil dianggap berbeda . Contoh-nya : variabel ‘A’ berbeda dengan variabel

"a" Untuk sintaks-sintaks dan fungsi-fungsi baku dalam MATLAB sebaiknya digunakan huruf kecil

7. Untuk melihat susunan fungsi-fungsi yang disediakan MATLAB dapat dilihat dengan menggunakan perintah HELP
8. Syntax penulisan >> help ENTER atau >> help *nama fungsi* <ENTER>

4. Aljabar MATRIKS

4.1. Penjumlahan dan pengurangan Matriks

Penjumlahan dan pengurangan matriks bisa dilakukan dengan syarat kedua matriks ber-ordo sama. Operasi dilakukan pada tiap-tiap elemen matriks yang sama.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix} \qquad \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a-e & b-f \\ c-g & d-h \end{pmatrix}$$

4.2. Perkalian Matriks

Perkalian dengan matriks dengan skalar bisa langsung dilakukan dengan mengalikan setiap elemen dengan nilai skalar.

$$k\begin{pmatrix} p & q \\ r & s \end{pmatrix} = \begin{pmatrix} kp & kq \\ kr & ks \end{pmatrix}$$

Perkalian matriks dengan matriks bisa dilakukan dengan syarat kolom matriks pertama sama dengan baris matriks kolom kedua. Misalkan A berordo pxq dan B berordo mxn, maka A X B jika q = m, hasil perkalian AB akan ber-ordo pxn

$$A = \begin{pmatrix} a & b & d \\ e & f & g \end{pmatrix}_{(2x3)}, \quad B = \begin{pmatrix} p & q \\ r & s \\ t & u \end{pmatrix}_{(3x2)}$$

$$A.B = \begin{pmatrix} a & b & d \\ e & f & g \end{pmatrix}_{(2x3)} \cdot \begin{pmatrix} p & q \\ r & s \\ t & u \end{pmatrix}_{(3x2)} = \begin{pmatrix} ap+br+dt & aq+bs+du \\ ep+fr+gt & eq+fs+gu \end{pmatrix}_{(2x2)}$$

4.3. Matriks Identitas

Matriks diagonal yang semua elemen diagonalnya adalah 1. Dengan sifat-sifat matriks identitas : A*I=A , I*A=A

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4.4. Determinan

Determinan adalah **nilai skalar yang dimiliki oleh sebuah matrik bujur sangkar**. *Nilai ini diperoleh sebagai hasil penjumlahan semua suku yang* dibentuk oleh permutasi elemen dari setiap vektor yang dapat dibentuk dari matrik tsb

4.5. Invers

Invers suatu matrik adalah matrik yang memenuhi definisi berikut:

Jika A = $[a_{ij}]$ dengan ordo nxn maka:

$A^{-1} = [a_{ij}]$ dengan ordo nxn dan memenuhi

$AA^{-1} = I$

$A^{-1}A = I$

5. PERSAMAAN LINIER

Bentuk persamaan linier dituliskan sebagai berikut:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3$$

....

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n$$

5.1. Metode Eliminasi Gauss

Eliminasi gauss digunakan untuk mencari akar sistem persamaan linier

Langkah penyelesaian:

1. Ubah sistem persamaan linier menjadi bentuk matriks (berordo nx(n+1))

   Berdasarkan bentuk persamaan linier point 5, maka bentuk matriksnya:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \cdots & a_{3n} & b_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & a_{nn} & b_n \end{pmatrix}$$

2. Melakukan reduksi baris di bawah diagonal utama sehingga nilainya adalah o (nol)

a. Periksa terlebih dahulu pivot/poros. Poros ≠ 0, jika bernilai 0 maka baris poros harus ditukar baris di bawahnya yang porosnya tidak nol
b. Pivot dimulai dari baris pertama kolom pertama
c. Lakukan reduksi baris pada baris berikutnya kolom pertama
d. Jika kolom pertama baris-baris di bawah baris poros sudah nol, maka cari poros berikutnya
e. Lakukan reduksi terus menerus sesuai dengan poros yang telah ditetapkan sampai nilai elemen di bawah diagonal utama menjadi nol (0)

3. Melakukan substitusi mundur (mulai dari baris paling bawah) untuk menentukan nilai variabel

5.2. Metode Eliminasi Gauss-Jordan

Eliminasi gauss digunakan untuk mencari akar sistem persamaan linier.

Langkah penyelesaian:

1. Ubah sistem persamaan linier menjadi bentuk matriks (berordo nx(n+1))
   Berdasarkan bentuk persamaan linier point 5, maka bentuk matriksnya:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & b_1 \\ 0 & 1 & 0 & \cdots & 0 & b_2 \\ 0 & 0 & 1 & \cdots & 0 & b_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & b_n \end{pmatrix}$$

2. Melakukan operasi baris elementer, artinya membuat nilai elemen di bawah dan di atas diagonal utama sehingga nilainya adalah o (nol)
   a. Periksa terlebih dahulu pivot/poros. Poros = 1;
      i. jika bernilai 0, maka baris poros harus ditukar baris di bawahnya yang porosnya tidak nol
      ii. jika bernilai >1,
         1. kalikan baris poros dengan $^1/_{poros}$
         2. kurangi baris poros dengan baris di bawahnya supaya poros bernilai 1 (satu)
      iii. jika bernilai < 0, maka kalikan dengan $^1/_{poros}$
   b. Pivot dimulai dari baris pertama kolom pertama
   c. Lakukan reduksi baris pada baris berikutnya kolom pertama
   d. Jika kolom pertama baris-baris di bawah baris poros sudah nol, maka cari poros berikutnya

e. Lakukan reduksi terus menerus sesuai dengan poros yang telah ditetapkan sampai nilai elemen di bawah diagonal utama menjadi nol (0)
f. Dimulai dari baris terakhir cari pivot/poros yang nilainya tidak nol
g. Lakukan reduksi baris-baris yang berada di atas pivot/poros supaya nilai elemennya menjadi nol (0) dan dikerjakan ke arah atas
h. Lakukan poin f s/d g sampai nilai di atas diagonal utama adalah nom (0)
i. Jika didapatkan matriks yang nilai elemen diagonalnya belum nol (0), maka lakukanlah perkalian matriks tersebut dengan matriks kolom dengan tujuan membuat nilai diagonal utama menjadi nol.

5.3. Metode Dekomposisi L-U

Dengan matriks kita bisa menyelesaikan persamaan linier diatas dengan bentuk Ax=b, sehingga bisa ditulis:

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

Penyelesaian Ax = b, dengan dekomposisi LU, maka:

1. Faktorkan A = LU, sehingga

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{3n} & \dots & a_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & 1 & 0 \\ \dots & \dots & \dots & & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & ua_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & \dots & u_{nn} \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & 1 & 0 \\ \dots & \dots & \dots & & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{pmatrix}$$

faktor pengali pada eliminasi Gauss diletakkan pada elemen yang bersesuaian di matriks L

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u a_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \cdots & \cdots & \cdots & & \cdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$ ➔ menggunakan teknik eliminasi Gauss

2. Jika

Ax = b

LUx = b, Misalkan Ux = y, maka Ly = b

3. Untuk memperoleh y, gunakan teknik substitusi maju (mulai dari baris paling atas)

$$Ly = b \rightarrow \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & 1 & \\ l_{n1} & l_{n2} & l_{n3} & & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_n \end{pmatrix}$$

4. Untuk memperoleh x, gunakan teknik substitusi mundur (mulai dari baris paling bawah)

$$Ux = y \rightarrow \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u a_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \cdots & \cdots & \cdots & & \cdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{pmatrix}$$

Operasi Matriks pada MATLAB

1. Mendefinisakan matriks

   Apabila kita ingin mendefinisikan sebuah matrik maka kita mengetikkan pada command window sebagai berikut : >> **A=[1 2 3 ; 4 5 6 ; 7 8 9]**

   Setelah mengetikan perintah tersebut kemudian kita menekan ↵ (enter) dan akan tampak hasil sebagai berikut :

   A =

   1 2 3

   4 5 6

   7 8 9

Membuat matrik dengan perulangan for maka kita mengetikkan algoritma-nya pada command window setelah pengetikan selesai diakhiri dengan end yang menyatakan akhir dari program

```
>> for i = 1:3,
       for j = 1:3,
           a(i,j) = 4*i-(3+j);
       end
   end
```

Program tersebut berarti didefinisikan i dari 1 sampai 3 yang merupakan baris dari matrik dan kemudian juga didefinisikan j dari 1 sampai 3 yang merupakan kolom matrik. Kemudian dibuat matrik a yang setiap elemen-nya merupakan hasil penambahan dari i dan j sesuai looping yag berjalan.

Untuk mengetahui hasilnya maka diketikkan a yang merupakan variable penampung hasil eksekusi program :

```
>> a
a =
    0   -1   -2
    4    3    2
    8    7    6
```

2. Operasi Penjumlahan

Inisialisasi matriks terlebih dahulu

```
>> A = [ 1 2 3 ; 4 5 6 ; 7 8 9]
A =
    1 2 3
    4 5 6
    7 8 9
```

```
>> B = [ 3 2 1 ; 6 5 4 ; 9 8 7]
B =
    3 2 1
    4 5 6
    7 7 9
```

Lalu berikan operasi penambahan:

```
>> A+B  <ENTER>
```

akan muncul hasilnya:

```
ans =
     4    4    4
     8   10   12
    14   15   18
```

3. Operasi Pengurangan

Dengan matriks A dan B yang sudah diidentifikasikan sebelumnya, lalukan operasi pengurangan

>> **A-B <ENTER>**

akan muncul hasilnya

ans =

-2 0 2

0 0 0

0 1 0

4. Operasi Perkalian

a. Perkalian dengan skalar

Definisikan skalar dan nilainya.

>> **k** = 2

>> **A*k**

ans =

1 4 6

8 10 12

14 16 18

b. Perkalian matriks dengan matriks

>> **A*B**

ans =

32 33 40

74 75 88

116 117 136

5. Mengubah nilai kolom dengan nilai tertentu

a. Definisikan dahulu matriks A

>> **A=[1 0 2; -3 4 6; -1 -2 3]**

A =

1 0 2

-3 4 6

-1 -2 3

b. Membuat matriks kolom b

>> **b =[6;30;8]**

b =

6

30

8

c. Mengganti kolom pertama matriks A dengan matriks kolom b

>> **A(:,1)=b**

A =

```
    6   0   2
   30   4   6
    8  -2   3
```

6. Mengubah nilai baris dengan nilai tertentu
   a. Definisikan dahulu matriks A

```
>> A=[1 0 2; -3 4 6; -1 -2 3]
A =
    1   0   2
   -3   4   6
   -1  -2   3
```

   b. Membuat matriks baris b

```
>> b =[6 30 8]
b =
    6   30   8
```

   c. Mengganti baris kedua matriks A dengan matriks baris b

```
>> A(2,:)=b
A =
    6   0   2
    6  30   8
    8  -2   3
```

Matlab untuk Menyelesaikan Persamaan Linier

1. **METODE GAUSS**

Selesaikan persamaan linier berikut:

$x + 2y + z = 3$

$3x - y - 3z = -1$

$2x + 3y + z = 4$

cara sederhana dengan MATLAB:

a. Buat matriks dari sistem persamaan yang ada

```
>> A=[1 2 1 3; 3 -1 3 -1; 2 3 1 4]
A =
    1   2   1   3
    3  -1   3  -1
    2   3   1   4
```

b. Reduksi baris kedua kolom pertama

```
>> A(2,:)=A(2,:)-3*A(1,:)
A =
```

```
    1   2   1   3
    0  -7   0  -10
    2   3   1   4
```

c. Reduksi baris ketiga kolom pertama

```
>> A(3,:)=A(3,:)-2*A(1,:)
A =
    1   2   1   3
    0  -7   0  -10
    0  -1  -1  -2
```

d. Reduksi baris ketiga kolom kedua

```
>> A(3,:)=A(3,:)-(1/7)*A(2,:)
A =
    1.0000   2.0000   1.0000   3.0000
         0  -7.0000  -6.0000  -10.0000
         0        0  -0.1429  -0.5714
```

e. Mencari nilai z, y dan x

```
>> z=A(3,4)/A(3,3)
z =
    4
>> y=(A(2,4)-(z*A(2,3)))/A(2,2)
y =
    -2
>> x=A(1,4)- (y*A(1,2))-(z*A(1,3))
x =
    3
```

2. **Metode GAUSS-JORDAN**

Selesaikan persamaan linier berikut:

x + 2y + z = 3

3x - y – 3z = -1

2x + 3y + z = 4

cara sederhana dengan MATLAB:

a. Buat matriks dari sistem persamaan yang ada

```
>> A=[1 2 1 3; 3 -1 3 -1; 2 3 1 4]
A =
    1   2   1   3
```

3 -1 3 -1
2 3 1 4

b. Reduksi baris kedua kolom pertama

```
>> A(2,:)=A(2,:)-3*A(1,:)
A =
     1     2     1     3
     0    -7     0   -10
     2     3     1     4
```

c. Reduksi baris ketiga kolom pertama

```
>> A(3,:)=A(3,:)-2*A(1,:)
A =
     1     2     1     3
     0    -7     0   -10
     0    -1    -1    -2
```

d. Reduksi baris ketiga kolom kedua

```
>> A(3,:)=A(3,:)-(1/7)*A(2,:)
A =
    1.0000    2.0000    1.0000    3.0000
         0   -7.0000   -6.0000  -10.0000
         0         0   -0.1429   -0.5714
```

e. Membuat nilai elemen diagonal menjadi nol (0)

```
>> A(2,:)=A(2,:)*(-1/7)
A =
    1.0000    2.0000    1.0000    3.0000
         0    1.0000    0.8571    1.4286
         0         0   -0.1429   -0.5714

>> A(3,:)=A(3,:)*(-1/0.1429)
A =
    1.0000    2.0000    1.0000    3.0000
         0    1.0000    0.8571    1.4286
         0         0    1.0000    3.9986
```

f. Mulai dari baris terbawah lakukan eliminasi untuk

```
>> A(2,:)=A(2,:)-(0.8571*A(3,:))
A =
```

```
    1.0000    2.0000    1.0000    3.0000
         0    1.0000         0   -1.9986
         0         0    1.0000    3.9986

>> A(1,:)=A(1,:)-(1*A(3,:))
A =
    1.0000    2.0000         0   -0.9986
         0    1.0000         0   -1.9986
         0         0    1.0000    3.9986

>> A(1,:)=A(1,:)-(2*A(2,:))
A =
    1.0000         0         0    2.9986
         0    1.0000         0   -1.9986
         0         0    1.0000    3.9986
```

g. Mencari nilai z, y dan x

```
>> x=A(1,4)
x =
   2.9986
>> y=A(2,4)
y =
   -1.9986

>> z=A(3,4)
z =
   3.9986
```

3. Metode DEKOMPOSISI LU

Tentukan solusi dari persamaan linier Ax=b, dimana:

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 2 & 2 & 1 \\ -1 & 1 & 1 \end{pmatrix} b = \begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix}$$

Penyelesaian sederhana dengan MATLAB:

a. Buat matriks A

```
>> A=[1 1 -1; 2 2 1; -1 1 1]
A =
    1    1   -1
    2    2    1
   -1    1    1
```

b. Buat matriks b

```
>> b=[1; 5; 1]
b =
     1
     5
     1
```

c. Mencari matriks L dan U

Gunakan fungsi yang sudah disediakan oleh MATLAB, yaitu fungsi **lu**

```
>> [L,U]=lu(A)
L =
    0.5000         0    1.0000
    1.0000         0         0
   -0.5000    1.0000         0

U =
    2.0000    2.0000    1.0000
         0    2.0000    1.5000
         0         0   -1.5000
```

d. Mencari solusi, yaitu nilai x

```
>> x=U\(L\b)
x =
     1
     1
     1
```


6. PRAKTIK MANDIRI

Kerjakan latihan berikut dengan MATLAB, cetak hasilnya dan berikan penjelasan dalam bentuk laporan tertulis/cetak. Dikumpulkan tanggal .... bulan ...... 2018)

1. Contoh penyelesaian masalah persamaan linier di atas (seperti dicontohkan sebelumnya) dengan menggunakan MATLAB adalah cara penyelesaian paling sederhana tanpa ada algoritma dan script program. Maka dari itu kembangkan sebuah algoritma program dan tulis script program yang lebih baik dengan MATLAB untuk persamaal linier berikut

$$x + 2y + z = 3$$
$$3x - y - 3z = -1$$