

IMPLEMENTASI PROGRAM SOFTWARE MATLAB DALAM MEMECAHKAN KASUS FISIKA: DINAMIKA SISTEM MASSA DAN PEGAS (PRINSIP NILAI DAN VEKTOR EIGEN)

Muh. Said L

Jurusan Fisika Fakultas Sains dan Teknologi
Universitas Islam Negeri Alauddin Makassar Kampus 2 Samata
e-mail: muhammadsaidlanto83@gmail.com

Abstract: Matlab Software Program Implementation in Solving Physics Problem: Dynamics Mass System and Spring (Values and Eigen Vectors Principles). Matlab software was applied in resolving physics problems especially in calculation the magnitudo and eigen vector. Physics problem solved in this article was oscillator motion which consists of four mass and four springs. Target of this program application was to determine the spring deviation length, thus the magnitudo as well as the eigen vector could be obtained. The method applied in this study was using the problem magnitudo and eigen vector to calculate spring deviation in Matlab. For the case of oscillatory motion, determination of each mass of system and the springs constants have been specified according to the system. Equation of motion for the mass-spring system was deduced by consider each point of masses use the function $q(t) = a_i e^{j\omega t}$, which is called as function of anzats, then derived twice respect to time. By using Matlab application, the result obtained indicate that by exploiting the eig (eigen) command in Matlab program, it was showed that running of the Matlab program would result accurately calculation for value and eigen vector.

Keywords: Matlab, dynamics, oscillator, magnitude and eigen vector

Abstrak: Implementasi Program Software Matlab dalam Memecahkan Kasus Fisika: Dinamika Sistem Massa dan Pegas (Prinsip Nilai dan Vektor Eigen). Software Matlab diaplikasikan dalam pemecahan kasus fisika menggunakan program perhitungan nilai dan vektor eigen. Kasus fisika yang dipecahkan adalah suatu benda yang bergerak secara osilator terdiri dari sistem empat massa dan empat pegas. Tujuan aplikasi program ini adalah untuk menentukan seberapa besar simpangan pegas, sehingga nilai (harga) dan vektor eigen dapat diperoleh. Metode yang dilakukan adalah memanfaatkan persoalan nilai dan vektor eigen untuk menghitung simpangan pegas dalam program Matlab. Pada kasus gerak osilator, penentuan besar masing-masing sistem massa dan konstanta pegas telah ditetapkan sesuai sistem. Persamaan gerak untuk sistem massa dan pegas masing-masing ditinjau setiap titik massa dan fungsi yang digunakan adalah $q(t) = a_i e^{j\omega t}$, yang disebut sebagai fungsi anzats, kemudian didiferensialkan dua kali terhadap waktu. Dengan menggunakan aplikasi program Matlab maka hasil yang diperoleh menunjukkan bahwa dengan memanfaatkan perintah eig (eigen) program matlab dapat menunjukkan secara perhitungan akurat hasil running (eksekusi) nilai dan vektor eigen..

Kata Kunci: Matlab, dinamika, osilator, nilai dan vektor eigen

Dalam kehidupan sehari-hari, banyak permasalahan dari fenomena real yang dapat dijelaskan melalui pembentukan model matematika. Pada umumnya perumusan model matematika ini berupa fungsi. Dalam banyak kasus, tidak semua model matematika tersebut dapat diselesaikan secara mudah dengan menggunakan metode analitik, sehingga digunakan metode numerik untuk mencari penyelesaiannya. Metode numerik adalah teknik yang digunakan untuk memformulasikan

persoalan matematik sehingga dapat dipecahkan dengan operasi perhitungan atau aritmetik biasa (tambah, kurang, kali, dan bagi).

Pada umumnya metode numerik tidak mengutamakan diperolehnya jawab yang eksak (tepat), tetapi mengusahakan perumusan metode yang menghasilkan jawab pendekatan yang berbeda dari jawab yang eksak sebesar suatu nilai yang merupakan galat dari metode yang digunakan. Namun demikian, hasil perhitungan

dengan metode numerik cukup dapat memberikan solusi pada persoalan yang dihadapi.

Salah satu penerapan dari metode numerik ini yaitu dalam masalah nilai eigen dan vektor eigen. Metode numerik memberikan suatu cara alternatif yang digunakan untuk menemukan nilai eigen dan vektor eigen dari suatu matriks. Cara yang digunakan dalam metode numerik ini termasuk unik karena dalam penyelesaiannya hanya diperlukan operasi-operasi aljabar biasa. Hanya saja, dalam penghitungannya tidak cukup dilakukan sekali tetapi harus dilakukan berulang-ulang sampai ditemukan nilai yang konvergen ke satu nilai yang merupakan nilai penyelesaiannya.

Nilai eigen banyak digunakan untuk mendapatkan solusi berbagai bidang. Karena permasalahan nilai eigen cukup penting kegunaannya, maka berbagai metode yang digunakan untuk menemukan nilai eigen menjadi penting untuk dipelajari. Metode numerik memberikan suatu cara alternatif yang digunakan untuk menemukan nilai eigen dan vektor eigen dari suatu matriks, salah satunya yaitu metode pangkat.

Dalam mencari nilai eigen dan vektor eigen menggunakan metode pangkat, akan memerlukan proses iterasi yang sangat panjang untuk menemukan hasil yang mendekati nilai yang sebenarnya. Semakin banyak iterasi yang dilakukan, maka semakin baik hasil yang diperoleh.

Rumusan masalah dari penelitian ini adalah:

Bagaimana menyelesaikan persoalan nilai dan vektor eigen dengan menggunakan aplikasi bahasa pemrograman Matlab?

Seberapa besar nilai dan vektor eigen pada proses pemecahan kasus dinamika (gerak osilator) sistem massa dan pegas dengan menggunakan aplikasi bahasa pemrograman Matlab?

TEORI

1. *Definisi Nilai dan Vektor Eigen*

Salah satu penerapan matriks pada persoalan fisika, dapat dijumpai dalam bentuk persamaan $Ax = \lambda x$. Dengan $A = a_{ij}$ adalah matriks bujur sangkar berorde (n) dan λ adalah suatu bilangan (skalar). Untuk solusi *trivial* $x = 0$ berapapun harga λ akan memenuhi, dan biasanya solusi ini tidak banyak gunanya dalam fisika. Untuk solusi *non-trivial* yaitu $x \neq 0$, harga λ yang memenuhi persamaan tersebut disebut *nilai eigen* atau nilai karakteristik dari matriks A dan solusi yang bersesuaian dengan persamaan yang diberikan $Ax = \lambda x$ disebut vektor eigen atau vektor karakteristik dari A.

Jika persamaan tersebut dinyatakan dalam bentuk sistem persamaan yang terpisah, misalnya:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n1} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n2} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

atau

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= \lambda x_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= \lambda x_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n &= \lambda x_3 \\ \cdot & \\ \cdot & \\ \cdot & \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= \lambda x_n \end{aligned}$$

Jika ruas kanan dipindahkan ke ruas kiri, maka persamaannya menjadi:

$$\begin{aligned} (a_{11} - \lambda)x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= 0 \\ a_{21}x_1 + (a_{22} - \lambda)x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= 0 \\ a_{31}x_1 + a_{32}x_2 + (a_{33} - \lambda)x_3 + \dots + a_{3n}x_n &= 0 \\ \cdot & \\ \cdot & \\ \cdot & \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + (a_{nn} - \lambda)x_n &= 0 \end{aligned}$$

2. Metode Jacobi

Salah satu metode untuk memecahkan persoalan nilai dan vektor eigen adalah metode *Jacobi*. Metode ini merupakan metode bentuk diagonalisasi suatu matriks dengan menggunakan sistem koordinat. Misal ditentukan koordinat dua dimensi sebagai berikut:

$$\begin{aligned} A_1 &= x_1 \cos\theta - x_2 \sin\theta \\ A_2 &= x_1 \sin\theta + x_2 \cos\theta \end{aligned} \tag{1}$$

Persamaan di atas dapat dinyatakan dalam bentuk matriks:

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2}$$

atau dapat diungkapkan dalam bentuk persamaan secara umum yaitu:

$$A = Tx \tag{3}$$

Matriks $T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ memiliki

matriks *transpose*:

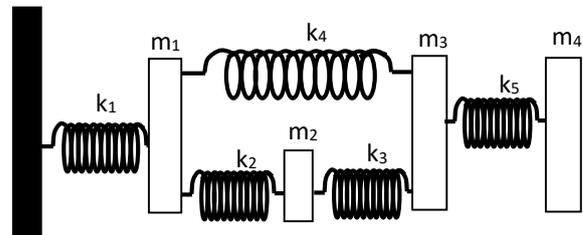
$$T^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{4}$$

Pada matriks T persamaan (2) dan T^T persamaan (4) dapat diperlihatkan sifat perkalian matriks yang disebut matriks diagonal. Matriks ini dapat digunakan untuk persamaan eigen.

$$T^T \cdot T = T \cdot T^T = I \text{ atau } T^T = T^{-1}$$

3. Kasus Fisika: Gerak Osilator Sistem Empat Massa dan Lima Pegas

Berikut satu kasus untuk memecahkan persamaan dinamika pada sistem massa dan pegas seperti gambar 1 berikut ini.



Gambar 1. Tinjauan sistem empat massa dan lima pegas.

Tinjauan pada setiap titik massa, diperoleh:

$$\begin{aligned} m_1 \ddot{q}_1 + (k_1 + k_2 + k_4)q_1 - k_2q_2 - k_4q_3 + (k_4 + k_3 + k_5)q_4 &= 0 \\ m_2 \ddot{q}_2 - k_2q_1 + (k_2 + k_3)q_2 - k_3q_3 + (k_3 + k_5)q_4 &= 0 \\ m_3 \ddot{q}_3 - k_4q_1 - k_3q_2 - (k_3 + k_4)q_3 - k_5q_4 &= 0 \\ m_4 \ddot{q}_4 + k_5q_4 &= 0 \end{aligned}$$

Fungsi *anzats* yang digunakan adalah:

$$q_i(t) = a_i \cdot e^{j\omega t} \tag{5}$$

dengan $i = 1, 2, 3$; dan $j =$ bilangan kompleks bernilai $\sqrt{-1}$, ω adalah simpangan pegas. Jika fungsi ini didiferensialkan dua kali terhadap waktu maka diperoleh:

$$\ddot{q}_i(t) = -\omega \cdot a_i \cdot e^{j\omega t} \tag{6}$$

Masukkan hasil diferensial tersebut ke masing-masing tinjauan setiap titik massa, sehingga dibentuk menjadi:

$$\begin{aligned} -\omega^2 m_1 a_1 + (k_1 + k_2 + k_4)a_1 - k_2 a_2 - k_4 a_3 + (k_4 + k_3 + k_5)a_4 &= 0 \\ -\omega^2 m_2 a_2 - k_2 a_1 + (k_2 + k_3)a_2 - k_3 a_3 + (k_3 + k_5)a_4 &= 0 \\ -\omega^2 m_3 a_3 - k_4 a_1 - k_3 a_2 + (k_3 + k_4)a_3 - k_5 a_4 &= 0 \\ -\omega^2 m_4 a_4 + k_5 a_4 &= 0 \end{aligned}$$

Pada kasus sistem massa dan pegas di atas, akan ditentukan masing-masing nilai konstanta pegas dan pegas yang digunakan yaitu:

$$m_1 = 10 \text{ kg}; m_2 = 20 \text{ kg}; m_3 = 30 \text{ kg}; m_4 = 40 \text{ kg};$$

$$k_1 = 10 \text{ kg.N/m}; k_2 = k_3 = 20 \text{ kg.N/m};$$

$$k_4 = 25 \text{ kg.N/m} \text{ dan } k_5 = 15 \text{ kg.N/m}$$

Masing-masing nilai di atas disubstitusikan ke persamaan tinjauan setiap titik massa:

$$-\omega^2 10a_1 + 55a_1 - 20a_2 - 25a_3 + 60a_4 = 0$$

$$-\omega^2 20a_2 - 20a_1 + 40a_2 - 20a_3 + 35a_4 = 0$$

$$-\omega^2 30a_3 - 25a_1 - 20a_2 + 45a_3 - 15a_4 = 0$$

$$-\omega^2 40a_4 + 15a_4 = 0$$

Dari empat tinjauan tersebut, maka dapat dinyatakan dalam bentuk:

$$-\omega^2 Aa + Ba = 0$$

Dengan

$$Aa = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix}$$

dan

$$Ba = \begin{bmatrix} 55 & -20 & -25 & 60 \\ -20 & 40 & -20 & 35 \\ -25 & -20 & 45 & -15 \\ 0 & 0 & 0 & 15 \end{bmatrix}$$

Untuk menentukan nilai simpangan pegas, dapat dihitung dengan menggunakan persoalan nilai dan vektor eigen. Nilai ω adalah vektor eigen.

METODE

A. Perangkat Lunak dan Keras yang digunakan

Dalam proses uji pemrograman komputer yang dilakukan dalam penelitian ini menggunakan perangkat lunak dan perangkat

keras. Proses uji coba dilakukan pada sebuah PC dengan intel (R) atom (TM) CPU N270 @ 1,60 GHz 798 MHz dengan memori 0,99 GB. Sistem operasi yang digunakan adalah Microsoft Windows XP Profesional SP2 version 2002 dan bahasa pemrograman yang telah digunakan adalah pada aplikasi metode yaitu MATLAB 6.5

B. Variabel Penelitian

Data parameter yang dibutuhkan terdiri dari beberapa variabel yaitu:

- Variabel bebas nilai massa dan konstanta pegas, kedua parameter ini sangat diperlukan untuk proses pemodelan dalam peninjauan gerak osilator dengan sistem empat massa dan lima pegas sesuai dengan kasus yang ditentukan seperti pada gambar 1.
- Variabel terikat: penentuan nilai dan vektor eigen dengan menggunakan aplikasi program Matlab.
- Variabel kontrol : fungsi yang digunakan dalam memecahkan kasus ini adalah fungsi *anzats*, kasus gerak osilator dengan sistem empat massa dan lima pegas yang ditentukan dan bahasa pemrograman yang difokuskan menggunakan Matlab.

C. Prosedur Penelitian

Dalam penelitian ini dilakukan langkah-langkah sebagai berikut:

- Membuat model kasus gerak osilator dengan sistem empat massa dan lima pegas seperti pada gambar 1 di atas, kemudian menetapkan masing-masing nilai massa dan konstantanya (nilai tidak boleh menggunakan data acak namun perlu diperhitungkan sesuai dengan tingkat keakuratan data)
- Meninjau setiap massa dan memodelkan kedalam bentuk persamaan diferensial orde dua.
- Menggunakan fungsi *anzats* $q_1(t) = a_i \cdot e^{j\omega t}$ sebagai fungsi pemecahan kasus dan mendiferensialkan dua kali terhadap waktu.

- d. Selanjutnya hasil diferensial tersebut ke masing-masing persamaan setiap massa (point b), selanjutnya mensubstitusi nilai massa dan konstanta pegas kemudian menghitungnya dan membuat model matriks sebagai langkah terakhir sebelum masuk dalam proses pembuatan program.
- e. Dari program yang telah dibuat maka selanjutnya me-*running* program tersebut dengan menginput semua elemen-elemen matriks tersebut, dan langkah selanjutnya mencetak kedua matriks yang pada akhirnya nilai dan vektor eigen diperoleh.

D. Algoritma Program

Langkah-langkah dalam membuat aplikasi program Matlab dalam penentuan nilai (harga) dan vektor eigen pada sistem empat massa dan lima pegas ditunjukkan dalam gambar 2 pada bagian akhir artikel ini.

HASIL DAN DISKUSI

A. Hasil Pemrograman dengan Komputasi

Sebelum proses pembuatan bahasa pemrograman, terlebih dahulu mendeskripsikan bentuk matriks yang akan diinput. Berikut hasil matriksnya adalah:

$$A = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix}$$

$$B = \begin{bmatrix} 55 & -20 & -25 & 60 \\ -20 & 40 & -20 & 35 \\ -25 & -20 & 45 & -15 \\ 0 & 0 & 0 & 15 \end{bmatrix}$$

Dengan melihat skema algoritma program seperti pada gambar 2, maka dibuat selanjutnya satu program komputasi menggunakan bahasa pemrograman dengan tujuan untuk menentukan nilai dan vektor eigen berdasarkan kasus fisika

berupa gerak osilator dengan sistem empat massa dan lima pegas, sehingga diperoleh hasil eksekusi nilainya. Adapun fase bahasa pemrograman komputasi dalam software Matlab terdiri dari beberapa langkah program yaitu:

1. Siapkan ruang untuk matriks A:

```
disp('Siapkan ruang untuk matriks A')
```

2. Masukkan jumlah baris dan kolom matriks A:

```
m=input('Masukkan jumlah baris=');
n=input('Masukkan jumlah kolom=');
M=zeros(m,n);
```

```
disp('')
```

3. Input elemen matriks A:

```
disp('## INPUT ELEMEN MATRIKS A ##')
for i=1:m
    for j=1:n
        disp(['A' num2str(i) ', ' num2str(j)'])
        A(i,j)=input('          = ');
    end
end
disp('')
disp('Matriks A')
disp('')
disp('A=')
disp(A)
disp('')
```

4. Siapkan ruang untuk matriks B:

```
disp('Siapkan ruang untuk matriks B')
```

5. Masukkan jumlah baris dan kolom matriks B:

```
x=input('Masukkan jumlah baris=');
y=input('Masukkan jumlah kolom=');
B=zeros(x,y);
disp('')
```

6. Input elemen matriks B:

```
disp('## INPUT ELEMEN MATRIKS B ##')
for i=1:x
    for j=1:y
        disp(['B' num2str(i) ', '
num2str(j)'])
        B(i,j)=input('          = ');
    end
end
disp('')
disp('Matriks B')
disp('')
disp('B=')
disp(B)
disp('')
```

7. Menentukan nilai eigen, $\lambda = \text{eig}(A)$:

```
disp('=====')
disp('==== Nilai Eigen ====')
disp('=====')
disp(' ')
lambda=eig(M,K)'
```

```
=0
A4,2
=0
A4,3
=0
A4,4
```

8. Menentukan vektor eigen, $\omega = \sqrt{\text{ones}(1,n)/\lambda}$:

```
disp(' ')
disp('=====')
disp('==== Nilai Eigen ====')
disp('=====')
disp(' ')
omega=sqrt(ones(1,y)./lambda)
```

```
=40
Matriks A
```

```
A =
    10     0     0     0
     0    20     0     0
     0     0    30     0
     0     0     0    40
```

Siapkan ruang untuk matriks B
Masukkan jumlah baris=4
Masukkan jumlah kolom=4

Berikut hasil *running* program dapat ditampilkan pada jendela *command window* setelah ditekan F5 sebagai berikut:

```
+++++
  PERSOALAN NILAI DAN VEKTOR
=====
+++++
Siapkan ruang untuk matriks A
Masukkan jumlah baris=4
Masukkan jumlah kolom=4

##INPUT ELEMEN MATRIKS A##
A1,1 =10
A1,2 =0
A1,3 =0
A1,4 =0
A2,1 =0
A2,2 =20
A2,3 =0
A2,4 =0
A3,1 =0
A3,2 =0
A3,3 =30
A3,4 =0
A4,1
```

```
## INPUT ELEMEN MATRIKS B ##
B1,1 = 55
B1,2 = -20
B1,3 = -25
B1,4 = 60
B2,1 = -20
B2,2 = 40
B2,3 = -20
B2,4 = 35
B3,1 = -25
B3,2 = -20
B3,3 = 45
B3,4 = -15
B4,1 = 0
B4,2 = 0
B4,3 = 0
B4,4 = 15

Matriks B
B=
    55   -20   -25    60
```

```

-20    40   -20    35
-25   -20    45   -15
  0     0     0    15
=====
====          Nilai Eigen          ====
=====

lambda =
  6.9542  0.1596  0.3862  2.6667

=====
====          Vektor Eigen          ====
=====

omega =
  0.3792  2.5034  1.6091  0.6124
    
```

Dari hasil penentuan dengan menggunakan aplikasi program dapat diketahui bahwa nilai eigen (*lambda*) diperoleh 0.0129; 0.0002; 0.0004; 0.0027 dan vektor eigen (*omega*) dihasilkan 8.7902; 76.3876; 50.8691; 19.3649.

Jadi dapat diperkuat bahwa nilai eigen merupakan harga *lambda* yang memenuhi karakteristik matematika-fisiknya adalah $Ax = \lambda x$, sedangkan vektor eigen merupakan solusi *x* yang bersesuaian dengan harga λ .

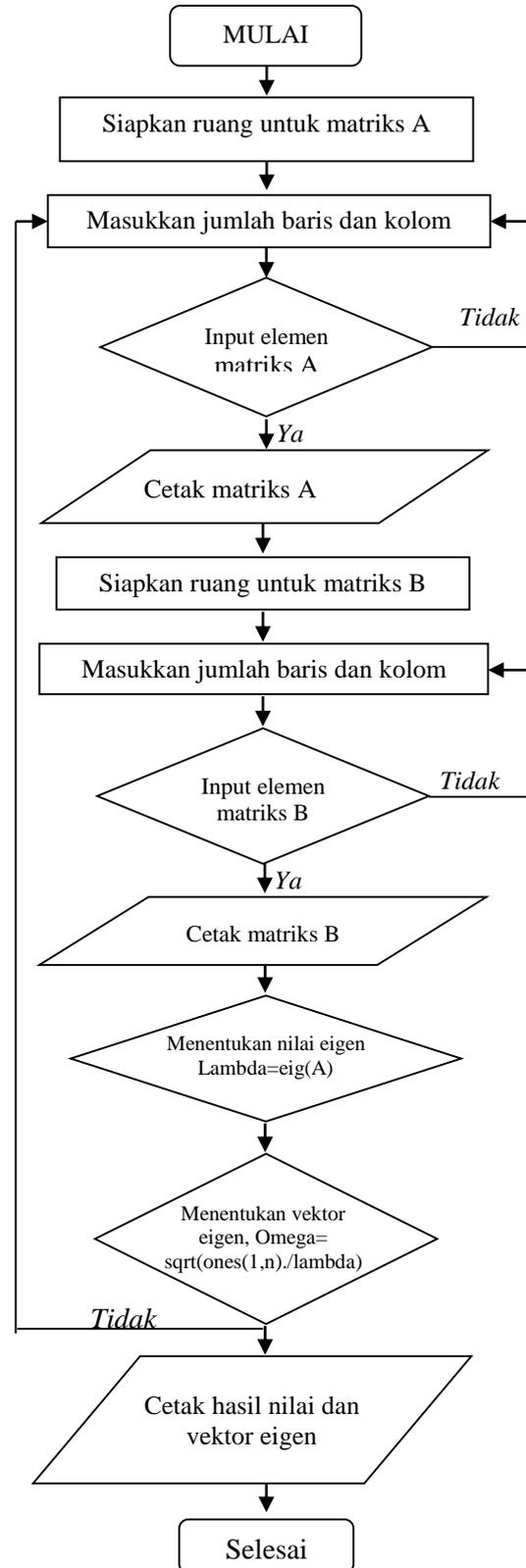
Penentuan nilai dan vektor eigen ini merupakan salah satu cara memecahkan sistem persamaan linier dalam kasus fisika, sehingga dapat disimpulkan secara jelas bahwa program software Matlab dapat menyelesaikan solusi kasus yang berkaitan dengan dinamika sistem massa dan pegas dalam konsep komputasi fisika matematika.

SIMPULAN

- a. Untuk menyelesaikan persoalan nilai dan vektor eigen dengan menggunakan aplikasi bahasa pemrograman Matlab dapat dilakukan dengan memanfaatkan perintah eig yaitu $\lambda = \text{eig}(A,B)$ untuk menentukan nilai eigen dan $\omega = \text{sqrt}(\text{ones}(1,y)/\lambda)$ untuk menentukan vektor eigen.
- b. Pada proses pemecahan kasus dinamika (gerak osilator) sistem massa dan pegas

dengan menggunakan aplikasi bahasa pemrograman Matlab diperoleh besar nilai eigen = 0.0129; 0.0002; 0.0004; 0.0027 dan vektor eigen 8.7902; 76.3876; 50.8691; 19.3649.

Algoritma program



DAFTAR RUJUKAN

- Aminuddin, Jamrud. 2008. *Dasar-Dasar Fisika Komputasi Menggunakan MATLAB*. Yogyakarta: Gava Media..
- Arfken, G., 1985. *Mathematical Methods for Physicists*. 3 rd ed., Academic Press.
- Boas, Mary, L. 1983. *Mathematical Methods in The Physical Sciences*. Secon Edition. John Wiley.
- Crosby, M.E., & Iding, M. K. 1997. *The Influence og a Multimedia Physics Tutor and User Diffrenceson the Development of Scientific Knowledge*. (pp.25-40). Computer & Education. 29, 127-136.
- Etter, D.M. 1993. *Engineering Problem Solving with MATLAB*. Prentice Hall. Englewood Cliff. New Jersey 07632.
- Giancolli, Douglas C., 2001. *Fisika*. Jilid 1. Ed. 5. Erlangga. Jakarta.
- Gunaidi, A.A., 2006. *MATLAB Programming*. Bandung: Informatika.
- Halliday, David & Robert Resnick. 1990. *Fisika Jilid 1 dan 2* (Alih bahasa: Pantur Silaban dan Erwin Sucipto), Edisi ke-3. Jakarta: Erlangga.
- Jimoyiannis, A.Mikropoulus,T. A. & Ravanis, K. 2000. *Students' Performance Towards Computer Simulations on Kinematics* (Themes In Education), 1 (4) (in press).
- Supriyanto Suparno, 2009, *Komputasi untuk Sains dan Teknik*. Edisi 3. Jakarta: Departemen Fisika-FMIPA, Univeristas Indonesia.
- Tao, P.K. 1997. *Confronting Student's Alternative Conceptions in Mechanics with the Force and Motion Microworld*. Computer in Physics 11 (2), 199-207.
- Roswati Mudjiarto, dkk. 2004. *Matematika Fisika 1*. Universitas Pendidikan Indonesia: Jurusan Pendidikan Fisika Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam.
- Schroeder , D.V., & Moore, T.A. 1993. A *Computer-Simulated Stern-Gerlach Laboratory*. American Journal of Physics. 61, 798-805.
- Wylie, C.R., Jr. 1961. *Advanced Engineering Mathematics*. McGraw-Hill, New York.