Klasifikasi Stabilitas Orbit Sistem Bintang dan Planet dengan Nilai Eigen

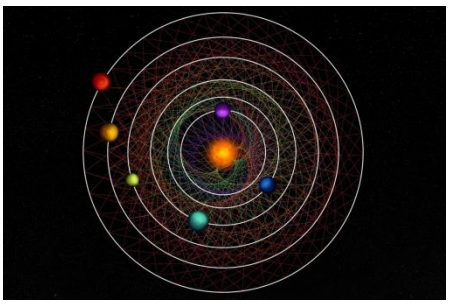
Muhammad Aditya Rahmadeni and 13532028[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523028@std.stei.itb.ac.id, [2]rahmadeniaditya@gmail.com

Abstrak **—Makalah ini membahas analisis stabilitas sistem orbit menggunakan metode linearisasi berbasis matriks Jacobian dan analisis nilai eigen. Penelitian ini bertujuan untuk memahami dinamika sistem orbit dan menentukan stabilitas orbit berdasarkan gangguan kecil yang terjadi dalam parameter awal. Dengan mendasarkan perhitungan pada persamaan gerak dua benda (two-body problem) dan metode Jacobian, orbit diuji dengan berbagai kondisi awal untuk mengklasifikasikan stabilitas berdasarkan bagian real dari nilai eigen..**

Keywords— **Stabilitas, Orbit, Matriks, Eigen**

I. PENDAHALUAN

Sistem bintang dan planet, atau sistem tata surya, bergerak secara melingkar antar satu sama lain atau yang dikenal dengan orbit. Studi orbit sangat penting dalam memahami dan memprediksi pergerakan benda langit, termasuk untuk satelit buatan dan pesawat ruang angkasa. Stabilitas orbit salah satu aspek penting dari studi orbit tersebut. Dengan mempelajari stabilitas orbit, Kerentanan benda langit dalam mempertahankan jalurnya dapat diketahui atau diprediksi. Stabilitas orbit dapat dipengaruhi oleh beberapa faktor, seperti gravitasi, gangguan dari objek langit lainnya, dan kondisi awal objek.



Gambar 1.1 Sistem Orbit Bintang dan Planet

Stabilitas orbit dapat dianalisis secara matematis menggunakan teori aljabar linear, khususnya nilai eigen dan matriks. Aljabar linear digunakan untuk memodelkan sistem dalam bentuk matriks yang merpresentasikan hubungan antara berbagai paramater dalam sistem.

Nilai eigen aalah bilangan skalar yang menunjukkan bagaimana suatu transformasi linear memengaruhi skalar vektor tertentu.. Dalam analisis stabilitas, nilai eigen dari matriks Jacobian, yang diperoleh dari linearisasi sistem non linier, memainkan peran penting. Nilai eigen dapat digunakan untuk menentukan apakah suatu sistem bersifat stabilitas, tidak stabil, atau besifat semi-stabil berdasarkan tanda bagian real dari nilail eigen tersebut.

Matrik Jacobi adalah matriks yang berasal dair linearisasi sistem nonlinier di sekitar titik ekuilibrium. Matriks Jacobi didefinisikan sebagai matriks yang elemen-lemennya adalah turunan paersial fingsi yang mendeskripsikam sistem terhadap variabel-variabelnya. Dengan kata lain, matriks Jacobian adalah representasi lokal dari dinamika sistem di sekitar titik tertentu

Makalah ini akan mengeksplorasi penggunaan dari matriks Jacobi dan nilai Eigen untuk menganalisis stabilitas orbit sistem bintang dan planet yang berfokus pada perhitungan dan interpretasi dari nilali eigen yang diturunkan dari matrix Jacobi sistem orbit itu sendiri.

II. LANDASAN TEORI

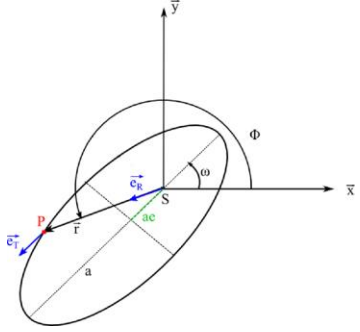
Pada bab ini akan dijelaskan landasan-landasan teori yang digunakan sesuai penjelasan singkat dari pendahuluan dan akan digunakan lebih lanjut pada bab selanjutnya,

1. Sistem Orbit

   Dalam menganalisis mekanisme dari sistem orbit, ada beberapa teorema yang digunakan sehingga sistem orbit dapat direpresentasikan dalam persamaan.

   a. Two-Body Problem

      Sistem orbit dalam konteks *two-body* didefinisikan dengan dua objek, yaitu badan utama, seperti bintang atau planet, dan badan sekunder, seperti planet atau satelit. Persamaan yang digunakan untuk menggamarkan teori ini, diturunkan dari hukum gravitasi Newton dan hukum gerak kedua Newton



Gambar 2.1Two-Body Problem Klasik

Hubungan posisi dan gerakan pada badan sekunder dimodelkan dalam:

- Vektor posisi ($\vec{r}$)
  Menggambarkan lokasi dari badan dalam sistem koodinat kartesius tiga dimensi (x, y, z)
- Vektor Kecepatan ($\vec{v}$)
  Menggambarkan perubahan posisi per satuan waktu.

Kedua komponen ini mementuk vektor kondisi($\vec{s}$)

$$\vec{s} = \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix}$$

b. Persamaan Gerak

Pergerakan benda dapat direpresentasikan dengan hukum newton, dengan menggabungkan konsep

- Gaya gravitasi
  Untuk menentukan percepatan akibat gravitasi
  $$F = -\frac{\mu \vec{r}}{r^2}$$
  Dengan r sebagai vektor posisi seperti yang telah dijelaskan sebelumnya
- Percepatan
  Diturunkan dari rumus gaya, yaitu
  $$F = m \cdot a$$
  dengan F adalah gaya dan m adalah massa serta a adalah percepatan itu sendiri. Sehingga bentuk percepatan sendiri memiliki bentuk
  $$\frac{d^2\vec{r}}{dt^2}$$

Dengan menggabungkan kedua konsep tersebut, maka dapat didapatkan

$$\frac{d^2\vec{r}}{dt^2} = -\frac{\mu \vec{r}}{r^3}$$

dengan r adalah panjang Euclidean dari $\vec{r}$ itu sendiri atau $\|\vec{r}\|$

c. Turunan

Setelah mendapatkan persamaan-persamaan diatas, turunan kedua dari persamaan diatas diubah menjadi turunan pertama untuk mempermudah perhitungan

$$\frac{d\vec{s}}{dt} = \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ -\frac{\mu \vec{r}}{r^3} \end{bmatrix}$$

2. Linearisasi

Lineariasai adalah proses mengubah sistem persaman turunan atau diferensial nonlinier ke dalam bentuk sistem persamaan turunan yang linier[1]. Matriks Jacobi akan diperoeh dari proses linierisasi ini. Dalam analisis kestabilan ini, penggunaan matriks Jacobi ini akan mempermudah untuk mencari aproksimasi linier dari persamaan sistem di sekitar sebuah titik dan memudahkan dalam menganalisis hal yang kompleks, seperti stabilitas, bifurkasi, dan tingkat perturbasi atau gangguan dari benda asing.

Secara matematis, matriks Jacobi adalah matriks setengah turunan atau turunan parsial yang merepresentasikan tingkat perubahan dari setiap fungsi persamaan dalam sebuah sistem terhadap sebuah variabel.

Misal sebuah fungsi vektor $\vec{f}$ memiliki variabel sebanyak n

$$\vec{f}(\vec{x}) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_m(x_1, x_2, \dots, x_n) \end{bmatrix}$$

Maka matriks Jacobi J akan memiliki bentuk

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Dalam bentuk umum,:

- $f_1, f_2, \dots, f_m$ adalah fungsi keluaran atau *output*
- $x_1, x_2, \dots, x_n$ adalah variabel yang akan menjadi input
- Setiap elemen pada matriks Jacobi atau $\frac{\partial f_i}{\partial x_i}$ adalah setengah turunan atau turunan parsial dari fungsi ke-i terhadap variabel ke-j

Dengan bentuk diatas, sistem orbit dapat direpresentasikan dengan sebagai berikut:

$$J = \begin{bmatrix} \frac{\partial \dot{\vec{r}}}{\partial \vec{r}} & \frac{\partial \dot{\vec{r}}}{\partial \vec{v}} \\ \frac{\partial \dot{\vec{v}}}{\partial \vec{r}} & \frac{\partial \dot{\vec{v}}}{\partial \vec{v}} \end{bmatrix}$$

dengan

- $\frac{\partial \dot{\vec{r}}}{\partial \vec{r}}$ adalah matriks nol
- $\frac{\partial \dot{\vec{r}}}{\partial \vec{v}}$ adalah matriks identitas
- $\frac{\partial \dot{\vec{v}}}{\partial r}$ adalah suku bukan nol yang diturunkan dari percepatan gravitasi
- $\frac{\partial \dot{\vec{v}}}{\partial \vec{v}}$ adalah matriks nol

Percepatan gravitasi diatas dapat diekspansi menjadi

$$\frac{\partial \dot{\vec{v}}}{\partial \vec{r}} = \mu\left(\frac{3\vec{r}\vec{r}^{T}}{r^5} - \frac{I}{r^3}\right)$$

Dengan I adalah matriks identitas

3. Nilai Eigen dan Vektor Eigen

Kata "eigen" berasal dari bahasa Jermain yang artinya "asli" atau "karakteristik".

Jika A adalah matriks n × n maka vektor tidak nol x di R^n disebut vektor Eigen dari A jika Ax sama dengan perkalian suatu skalar λ dengan x, yaitu

$$Ax = \lambda x$$

Skalar λ disebut nilai Eigen dari A, dan x dinamakan vektror eigen yang berkorespoden dengan λ[4].

Vektor Eigen adalah matriks kolom yang jika dikalikan dengan sebuah matriks persegi akan menghasilkan vektor lain yang merupakan kelipatan vektor itu sendiri.

Sehingga, operasi diatas disimpulkan menjadi vektor x menyusut atau memanjang dengan faktor λ dengan arah yang sama dengan jika bernilai positif dan berkebalikan arah jika bernilai negatif.

Persamaan diatas dapat disusun kembali menjadi persamaan

$$(\lambda I - A)\, x = 0$$

Dengan I adalah matriks identitas berukuran yang sama dengan matriks A.

Agar λ memiliki nilai, maka matriks $\lambda I - A$ haruslah memiliki solusi tak nol atau memiliki solusi non-trivial, sehingga λ dapat dicari dengan

$$|\lambda I - A| = 0$$

Atau determinan dari matriks diatas haruslah bernilai 0.

4. Analisis Stabilitas

Analisis stabilitas adalah hal penting dalam topik ini Dengan menggunakan matriks Jacobi, stablitas di titiik ekuilibrium dan jalur dari sistem dapat dianalisis.

Persamaan dinamis sistem secara umum dapat digambarkan sebagai

$$\dot{\vec{x}} = \vec{f}(\vec{x})$$

Dimana x adalah vektor kondisi berdimensi n dan fungsi f adalah vektor non-linier yang menggambarkan evolusi atau pergerakan sistem secara melingkar dari sistem per waktu.

Titik ekuilibrium x adalah titik dimana kecepatan sistem menjadi nol

$$\vec{f}(\vec{x}^*) = 0$$

Dengan definisi titik ekuilibrium diatas, kestabilan orbit dapat ditentukan dengan tiga kondisi

- Stabil
  Sistem stabil kika setiap kondisi awal dekat dengan titik ekuilibrium dan kondisi tetap dekat dengan titik ekuilibrium tersebut selama $t \to \infty$
- Stabil Asimptotik
  Sistem akan stabil secara asimptotik jika kondisi konvergen ke tititk ekuilibrium selama $t \to \infty$
- Tidak Stabil
  Sistem tidak stabil jika perturbasi atau gangguan meningkat terus menerus

.Sistem dapat dilinierisas disekitar titik ekuilibrium dengan matriks Jacobi

$$\dot{\vec{x}} = J(\vec{x}^*)\partial\vec{x}$$

dengan $\partial\vec{x} = \vec{x} - \vec{x}^*$ melambangkan penyimpangan kecil dari ekuilibrium.

Dari matriks Jacobi atau matriks hasil linierisasi sistem, nilai eigen dapat ditentukan dan kestabilan orbit dapat ditentukan baik jika hasilnya berupa bilangan riil atau imajiner.

- Analisis bilangan rill:
  Jika semua nilai eigen bernilai negatif untuk bagian rill nya, maka sistem stabil secara asimptotik dan perturbasi akan menghilang seiring waktu
  Jika ada nilai eigen yang mempunyai nilai riil yang positif, maka sistem tidak stabil dan perturbasi atau gangguan akan meningkat secara eksponensial
  Jika nilai eigen mempunyai nilai riil 0, maka stabilitas tidak dapat diukur dari linierisasi saja dan memperlukan analisis tingkat tinggi
- Analisis bilangan imajiner
  Nilai eigen yang hanya memiliki nilai imajiner mengindikasikan perilaku osilasi atau getar dalam orbit. Jika nilai rill adalah nol, maka sistem sedikit stabil yang berarti getaran atau osilasi tidak meningkat ataupun berkurang.

III. IMPLEMENTASI

Untuk membuktikan dari konsep dan teori yang telah dijelaskan pada bab landasan teori, implementasi dilakukan sehingga percobaan dapat dilakukan secara langsung.

Program akan dibuat menggunakan bahasa pemograman Python dengan alasan untuk mempermudah karena pilihan sintaks dan modul yang telah disediakan.

Setiap perhitungan akan dibagi menjadi fungsi yang berbeda untuk mempermudah dalam proses *debugging* dan dengan alasan mempermudah penjelasan.

Perhitungan nilai eigen dan vektor eigen dilakukan dengan menggunakan modul *scipy*. Sehingga dilakukan impor modul terlebih dahulu

```
from scipy.integrate import solve_ivp
from scipy.linalg import eig
```

Selain itu, modul dari numpy dan matplotlib digunakan untuk perhitungan yang lebih cepat dan akurat serta visualisasi dari hasil yang didapat untuk membuktikan kestabilan dari orbit.

```
import numpy as np
import matplotlib.pyplot as plt
```

Disini tubuh utama akan menggunakan matahari sehingga dibuat variabel global untuk menympan massa dan nilai gravitasi dari matahari

```
G = 6.67430e-11
M_sun = 1.989e30
mu = G * M_sun
```

Untuk menghitung turunan persamaan dinamis dari sistem orbit, dibuat sebuah fungsi dengan alasan modularitas. Teori gravitasi Newton diaplikasikan sehingga terbentuk fungsi

```
def orbitalEquation(t, y):
    x, y, z, vx, vy, vz = y
    r = np.sqrt(x**2 + y**2 + z**2)
    ax = -mu * x / r**3
    ay = -mu * y / r**3
    az = -mu * z / r**3
    return [vx, vy, vz, ax, ay, az]
```

Dengan persamaan tersebut, dibentuk matriks Jacobi yang akan melinierisasi persamaan yang telah dibentuk sebelumnya menggunaakan fungsi diatas. Semua turunan dari setiap elemen pada matriks Jacobi akan dihitung secara manual

```
def calculate_jacobian(r, v):
    x, y, z = r
    r_norm = np.linalg.norm(r)
    d2x_dx = -mu / r_norm**3 + 3 *
mu * x**2 / r_norm**5
    d2x_dy = 3 * mu * x * y /
r_norm**5
    d2x_dz = 3 * mu * x * z /
r_norm**5
    d2y_dx = d2x_dy
    d2y_dy = -mu / r_norm**3 + 3 *
mu * y**2 / r_norm**5
    d2y_dz = 3 * mu * y * z /
r_norm**5
    d2z_dx = d2x_dz
    d2z_dy = d2y_dz
    d2z_dz = -mu / r_norm**3 + 3 *
mu * z**2 / r_norm**5
    jacobian = np.zeros((6, 6))
    jacobian[:3, 3:] = np.eye(3)
    jacobian[3:, :3] = np.array([
        [d2x_dx, d2x_dy, d2x_dz],
        [d2y_dx, d2y_dy, d2y_dz],
        [d2z_dx, d2z_dy, d2z_dz]
    ])
    return jacobian
```

Fungsi untuk mengkalkulasi semua hal yang dibutuhkan telah didapatkan. Selanjutnya dilakukan inisialisasi kondisi untuk simulasi dan perhitungan

```
r0 = np.array([1.496e11, 0, 0])
v0 = np.array([0, 29.78e3, 0])
y0 = np.hstack((r0, v0))


t_span = (0, 3.154e7)
t_eval = np.linspace(t_span[0],
t_span[1], 1000)
```

Pada blok kode ini, dilakukan inisilaisasi posisi awal dan kecepatan awal serta waktu akhir perhitungan selama satu tahun. Waktu disini memilki makna perkiraan atau prediksi pergerakan orbit tersebut selama satu tahun.

Lalu, dilakukan perhitngan nilai eigen dari matriks Jacobi yang telah dihitung sebelumnya yang nantinya digunakan untuk menentukan stabilitas orbit tersebut.

```
jacobian = calculateJacobi(r0, v0)
eigenvalues, eigenvectors =
eig(jacobian)
stability = all(e.real <= 0 for e in
eigenvalues)
print(f"Eigenvalues: {eigenvalues}")
print(f"Is the system stable? {'Yes'
if stability else 'No'}")
```

Pada kondisi ini, program telah dapat menentukan apakah orbit stabil atau tidak. Untuk mempermudah dalam membuktikan apakah perhitungan benar atau tidak, dilakukan visualisasi menggunakan modul matplotlib yang telah diimpor sebelumnya

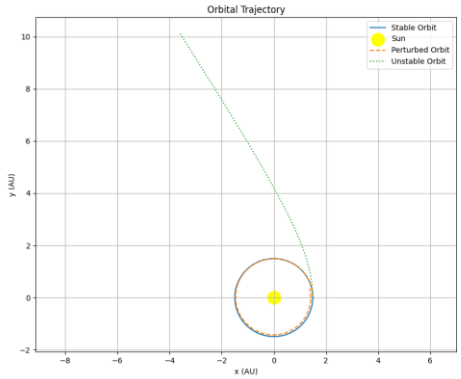
```
plt.figure(figsize=(10, 8))
plt.plot(x / 1e11, y / 1e11, label="Stable Orbit")
plt.scatter([0], [0], color='yellow', s=300,
label="Sun")
plt.title("Orbital Trajectory")
plt.xlabel("x (AU)")
plt.ylabel("y (AU)")
plt.legend()
plt.grid()
plt.axis("equal")
```

Sehingga didapatkan kode program secara utuh seperti dibawah ini

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from scipy.linalg import eig


G = 6.67430e-11
M_sun = 1.989e30
mu = G * M_sun

def orbitalEquation(t, y):
    x, y, z, vx, vy, vz = y
    r = np.sqrt(x**2 + y**2 + z**2)
    ax = -mu * x / r**3
    ay = -mu * y / r**3
    az = -mu * z / r**3
    return [vx, vy, vz, ax, ay, az]

def calculateJacobi(r, v):
    x, y, z = r
    r_norm = np.linalg.norm(r)
    d2x_dx = -mu / r_norm**3 + 3 * mu
* x**2 / r_norm**5
    d2x_dy = 3 * mu * x * y /
r_norm**5
    d2x_dz = 3 * mu * x * z /
r_norm**5
    d2y_dx = d2x_dy
    d2y_dy = -mu / r_norm**3 + 3 * mu
* y**2 / r_norm**5
    d2y_dz = 3 * mu * y * z /
r_norm**5
    d2z_dx = d2x_dz
    d2z_dy = d2y_dz
    d2z_dz = -mu / r_norm**3 + 3 * mu
* z**2 / r_norm**5
    jacobian = np.zeros((6, 6))
    jacobian[:3, 3:] = np.eye(3)
    jacobian[3:, :3] = np.array([
        [d2x_dx, d2x_dy, d2x_dz],
        [d2y_dx, d2y_dy, d2y_dz],
        [d2z_dx, d2z_dy, d2z_dz]
    ])
    return jacobian


r0 = np.array([1.496e11, 0, 0])
v0 = np.array([0, 29.78e3, 0])
y0 = np.hstack((r0, v0))


t_span = (0, 3.154e7)
t_eval = np.linspace(t_span[0],
t_span[1], 1000)


solution = solve_ivp(orbitalquation,
t_span, y0, t_eval=t_eval, rtol=1e-9,
atol=1e-12)
x, y, z = solution.y[:3]
```

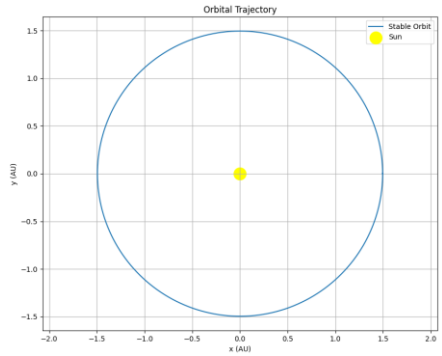
```
jacobian = calculateJacobi(r0, v0)
eigenvalues, eigenvectors =
eig(jacobian)
stability = all(e.real <= 0 for e in
eigenvalues)
print(f"Eigenvalues: {eigenvalues}")
print(f"Is the system stable? {'Yes'
if stability else 'No'}")


plt.figure(figsize=(10, 8))
plt.plot(x / 1e11, y / 1e11,
label="Stable Orbit")
plt.scatter([0], [0], color='yellow',
s=300, label="Sun")
plt.title("Orbital Trajectory")
plt.xlabel("x (AU)")
plt.ylabel("y (AU)")
plt.legend()
plt.grid()
plt.axis("equal")
```

Dilakukan pengetesan kode ini dan berikut hasil yang didapatkan serta penambahaan kondisi dimana jika adanya perturbasi dan jika kecepatan orbit ditambah dengan kecepatan yang tidak normal($0x5 \times 10^3$) untuk menggambarkan ketidakstabliian.



Gambar 3.1 Hasil perhtiungan

Sebagai pembanding berikut adalah hasil orbit stabil yang mana pada contoh kali ini orbitt yang digunakan adalah orbit bumi dan matahari



Gambar 3.2 Hasil perhitungan orbit stabil

IV. CONCLUSION

Berdasarkan analisis dan implementasi yang dilakukan, dapat disimpulkan bahwa matriks Jacobian dan analisis nilai eigen merupakan alat yang efektif untuk mengevaluasi stabilitas sistem orbit. Dengan menggunakan linearisasi persamaan gerak, dinamika orbit dapat direpresentasikan dalam bentuk matriks yang memberikan informasi mengenai sensitivitas sistem terhadap gangguan kecil. Analisis menunjukkan bahwa orbit stabil jika semua nilai eigen memiliki bagian real negatif, sementara orbit menjadi tidak stabil jika terdapat nilai eigen dengan bagian real positif.

Namun, metode ini memiliki keterbatasan karena mengasumsikan sistem dua benda yang ideal dan linier dan mengabaikan pengaruh eksternal seperti interaksi gravitasi dengan benda-benda lain, efek relativitas umum, atau tekanan radiasi. Oleh karena itu, analisis stabilitas yang lebih kompleks diperlukan untuk menggambarkan dinamika sistem orbit yang lebih realistis. Implementasi Python yang dikembangkan dalam makalah ini dapat digunakan sebagai alat simulasi untuk mempelajari stabilitas orbit dan memberikan landasan bagi pengembangan analisis yang lebih lanjut.

REFERENCES

[1] Putri, E. S. (2018). *Penerapan Matriks Jacobian untuk Analisis Stabilitas Orbit*. Universitas Negeri Yogyakarta. Diakses dari https://eprints.uny.ac.id/54892/

[2] Vinh, N. X. (1982). *Jacobian Stability in Orbital Mechanics*. NASA Technical Reports Server. Diakses dari https://ntrs.nasa.gov/citations/19820048831

[3] SpringerLink. (2024). *Eigenvalues and Orbital Stability in Astrophysics*. Astrophysics and Space Science. Diakses dari https://link.springer.com/article/10.1007/s10509-024-04312-8

[4] Munir, R. (2023). *Nilai Eigen dan Vektor Eigen: Bagian 1*. Institut Teknologi Bandung. Diakses dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian1-2023.pdf

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 2 Januari 2024

Muhammad Aditya Rahmadeni 135230288