p-ISSN:2527-5321
e-ISSN:2527-5941

SOLUSI SIMBOLIS DAN VISUALISASI PENDULUM PEGAS MENGGUNAKAN LIBRARY SYMPY DI PYTHON

Didik Nur Huda[1], Santy Handayani[2]

Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI Jakarta
Jalan Raya Tengah Kelurahan Gedong, Pasar Rebo, Jakarta Timur 13760
[1]didiks.physics@gmail.com, [2] santyhandayani1@gmail.com

ABSTRAK

Pendulum pegas merupakan sistem mekanika klasik. Solusi persamaan pendulum pegas akan rumit jika tanpa perangkat komputasi. Persamaan Euler-Lagrange yang mengendalikan gerak pendulum pegas merupakan sistem persamaan diferensial orde dua. Penyelesaian Persamaan Euler-Lagrange sangat kompleks jika dilakukan manual dengan tulisan tangan. Solusi persamaan Euler-Lagrange secara simbolis dapat menggunakan *Library* SymPy 1.6.2 di Phyton dibandingkan dengan analisis manual. Solusi tersebut selanjutnya dibuat animasi untuk visualisasi gerak pendulum pegas. Penurunan dengan Library SymPy sudah cocok dengan penurunan manual yaitu $m\ddot{x} - m(l+x)\dot{\theta}^2 - mg\cos\theta + kx$ dan $m(l+x)\ddot{\theta} + 2m\dot{x}\dot{\theta} + mg\sin\theta$, serta visualisasi juga sudah sesuai harapan.
Kata Kunci: Persamaan Diferensial, Solusi Persamaan Diferensial, SymPy, Python

ABSTRACT

The spring pendulum is a classical mechanical system. The solutions to the equations of a spring pendulum become complex without computational tools. The Euler-Lagrange equation that controls the motion of the spring pendulum is a system of second order differential equations. The solution of Euler-Lagrange equation is highly comolex when done manually with handwritten calculation. Symbolic solutions of the Euler-Lagrange equations using SymPy 1.6.2 in Python are compared to manual analysis. The solution is then made into an animation to visualize the motion of the spring pendulum. The dervation using the SymPy library is compatible with manual derivation which is $m\ddot{x} - m(l+x)\dot{\theta}^2 - mg\cos\theta + kx$ *and* $m(l+x)\ddot{\theta} + 2m\dot{x}\dot{\theta} + mg\sin\theta$, *and the visualizations also as expected.*
Key Word: *Differential Equations, Differential Equation Solutions, SymPy, Python.*

PENDAHULUAN

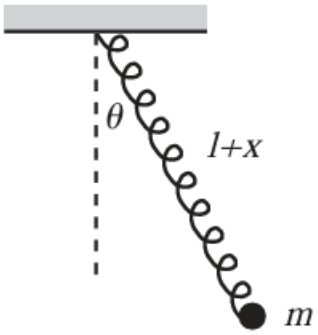
Pendulum pegas merupakan salah satu sistem mekanika klasik yang memiliki aplikasi luas dalam berbagai bidang. Sistem ini terdiri dari sebuah massa yang dihubungkan ke pegas, membentuk suatu sistem yang dapat berayun atau bergetar (Gambar 1). Solusi anatilik dari gerakan pendulum pegas seringkali melibatkan penyelesaian persamaan diferensial orde dua, yang dapat menjadi tugas yang rumit tanpa bantuan perangkat lunak komputasi (Boas, 1983). Solusi analitik dalam buku *Essential Mathematical Method fo Phycisists* dapat dikerjakan dengan bantuan perangkat lunak komputer (Weber & Arfken, 2004). Solusi menggunakan mekanika Lagrange dapat digunakan untuk menyelesaikan pendulum pegas ini (Wells, 1967). Selain menggunakan mekanika Lagrange dapat juga menggunakan mekanika Newton untuk mencari penyelesaian pendulum pegas ini dan dapat membuat animasinya (Neumann, 2021).

Dalam penyelesaiannya Mekanika *Lagrange* dan Mekanika Newton membutuhkan matematika yang cukup kompleks jika ditulis dengan tangan (Broucke & Baxa, 1973). Tidak hanya menggunakan dua mekanika di atas, ada juga yang menggunakan Mekanika Hamilton untuk menyelesaikan masalah energi pada pendulum pegas (De Sousa et al., 2017). Ada juga dengan model fraksional untuk menyelesaikan persamaan pendulum pegas (Baleanu et al., 2018). Ternyata pendulum pegas dapat dimanfaatkan untuk memanen energi (Abohamer et al., 2021).

Python menjadi salah satu bahasa pemrograman yang paling banyak digunakan untuk simulasi fisika dan analisis matematis. Library SymPy merupakan salah satu alat yang kuat dalam ekosistem Python, yang menyediakan solusi simbolis untuk persamaan diferensial dan algebra (*Solve an Equation Algebraically*, 2023). Dengan memanfaatkan Library SymPy dapat

p-ISSN:2527-5321
e-ISSN:2527-5941

mengatasi tingkat kompleksitas dalam penyelesaian persamaan gerak pendulum pegas.



Gambar 1. Pendulum pegas (*spring pendulum*)

Artikel ini bertujuan untuk melihat kemampuan Libary SymPy dalam menyelesaikan persamaan diferensial pada mekanika Lagrange secara simbolis. Selain itu, membuat animasi sederhana di Python untuk mengilustrasikan perubahan dinamika sistem seiring waktu. Dengan menggabungkan solusi simbolis dan visualisasi animatif, diharapkan akan mendapatkan pemahaman yang mendalam tentang perilaku kompleks dari pendulum pegas.

METODE PENELITIAN

Penelitian ini dilakukan dimana saja, hanya berbekal laptop dan jaringan internet. Data berupa ungkapan persamaan Euler-Lagrange untuk pendulum pegas sebagai berikut (Morin, 2008):

Energi kinetik:

$$E_k = \frac{1}{2}m(\dot{x}^2 + (l+x)^2\dot{\theta}^2) \quad (1)$$

Energi potensial:

$$E_p = -mg(l+x)\cos\theta + \frac{1}{2}kx^2 \quad (2)$$

Lagrangian:

$$L \equiv E_k - E_p \quad (3)$$

Persamaan Euler-Lagrange sebagai berikut:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0 \quad (4)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \quad (5)$$

Dari persamaan E_k dan E_p diturunkan dengan Library SymPy di Python dengan JupyterLab IDE untuk menghasilkan solusi persamaan Euler-Lagrange. Bandingkan solusi dari SymPy dengan buku (Morin, 2008). Selanjutnya dari solusi persamaan Euler-Lagrange dibuat animasi sederhana. Bandingkan animasi dari Python dengan sumber (Neumann, 2021) dengan metode Euler-Lagrange.

HASIL DAN PEMBAHASAN

Penurunan perumusan persamaan Euler-Lagrange dengan Library SymPy 1.6.2 di Python 3.9 dengan JupyterLab IDE adalah sebagai berikut:

Penurunan terhadap $\dot{x}$

$$-gm\cos(\theta(t)) + 1.0kx(t) - 1.0m(l+x(t))\left(\frac{d}{dt}\theta(t)\right)^2 + 1.0m\frac{d^2}{dt^2}x(t)$$

Penurunan terhadap $\dot{\theta}$

$$m(l+x(t))\left(g\sin(\theta(t)) + 1.0(l+x(t))\frac{d^2}{dt^2}\theta(t) + 2.0\frac{d}{dt}\theta(t)\frac{d}{dt}x(t)\right)$$

Alasan utama menggunakan JupyterLab karena dapat menampilkan persamaan matematika dengan baik dan bagus.

Jika dibandingkan dengan sumber

Penurunan terhadap $\dot{x}$

$$m\ddot{x} - m(l+x)\dot{\theta}^2 - mg\cos\theta + kx$$

Penurunan terhadap $\dot{\theta}$

$$m(l+x)\ddot{\theta} + 2m\dot{x}\dot{\theta} + mg\sin\theta$$

Dari hasil SymPy perlu sedikit operasi aljabar untuk menyesuaikan dengan sumber dan perlu diingat bahwa simbol titik di atas huruf (∴) untuk simbol diferensial sekali terhadap waktu dan simbol dua titik di atas huruf (⁖) untuk diferensial dua kali terhadap waktu.

Terlihat bahwa persamaan Euler-Lagrange merupakan persamaan diferensial biasa orde dua. Solusi untuk dua persamaannya adalah sebagai berikut:

$$\frac{1.0gm\cos(\theta(t)) - 1.0kx(t) + 1.0lm\left(\frac{d}{dt}\theta(t)\right)^2 + 1.0mx(t)\left(\frac{d}{dt}\theta(t)\right)^2}{m}$$

dan

$$\frac{-1.0glm\sin(\theta(t)) - 1.0gmx(t)\sin(\theta(t)) - 2.0lm\frac{d}{dt}\theta(t)\frac{d}{dt}x(t) - 2.0mx(t)\frac{d}{dt}\theta(t)\frac{d}{dt}x(t)}{l^2m + 2lmx(t) + mx^2(t)}$$

SymPy juga mendukung untuk penyelesaian numerik dari solusi simbolisnya secara langsung. Untuk memperoleh persamaan numeriknya dengan lambdify.

```
dz1dt_f = smp.lambdify((t,g,m,k,l,x,the,x_d,the_d), sols[x_dd])
dz2dt_f = smp.lambdify((t,g,m,k,l,x,the,x_d,the_d), sols[the_dd])
dxdt_f = smp.lambdify(x_d, x_d)
dthedt_f = smp.lambdify(the_d, the_d)
```

Gambar 2. Mengubah simbolis menjadi variabel numerik

Agar dapat diselesaikan dengan metode numerik bentuk persamaanya perlu diubah.

p-ISSN:2527-5321
e-ISSN:2527-5941

Karena SymPy hanya mendukung penyelesaian persamaan diferensial biasa orde satu maka perlu diubah bentuknya menjadi:

$$z_1 = \frac{dx}{dt}$$
$$z_2 = \frac{d\theta}{dt}$$

Setelah diubah menjadi variabel z_1 dan z_2 dibuatlah suatu fungsi $S(x, z_1, \theta, z_2)$ yang dinyatakan sebagai vektor dalam NumPy. NumPy dibutuhkan untuk penyelesaian numerik ini.

```
def dSdt(S, t, g, m, l, k):
    x, z1, the, z2 = S
    return [
        dxdt_f(z1),
        dz1dt_f(t, g, m, l, k, x, the, z1, z2),
        dthedt_f(z2),
        dz2dt_f(t, g, m, l, k, x, the, z1, z2),
    ]
```

Gambar 3. Fungsi *S*

dengan

$\text{dxdt\_f(z1)} = \frac{dx}{dt} = z_1$

$\text{dz1dt\_f(...)} = \frac{dz_1}{dt}$

$\text{dthedt\_f(z2)} = \frac{d\theta}{dt} = z_2$

$\text{dz2dt\_f(...)} = \frac{dz_2}{dt}$

Selanjutnya dengan mensubstitusi variabel dan syarat awal ke fungsi $S(x, z_1, \theta, z_2)$ akan diperoleh data yang digunakan untuk membuat animasi.

Berikut konstanta-konstanta dan syarat awal yang disubstitusikan:

$t = [0, 100]$
$g = 9{,}81$
$m = 0{,}5$
$l = 2{,}5$
$k = 6$
$x_0 = 2$
$z_1 = 0$
$\theta = 2$
$z_2 = 0$

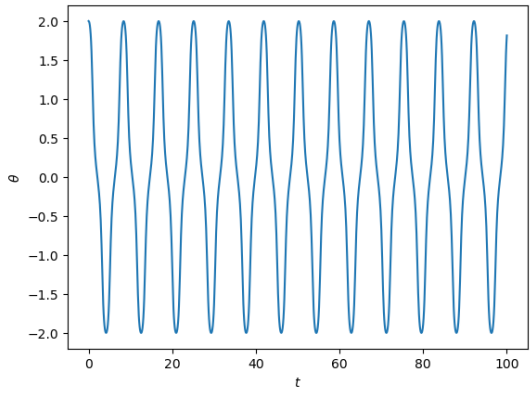
```
t = np.linspace(0, 100, 1001)
g = 9.81
m = 0.5
l = 2.5
k = 6
ans = odeint(dSdt, y0=[2, 0, 2, 0], t=t, args=(g,m,l,k)) #y0=[x, z1, the, z2]
```

Gambar 4. Konstanta-konstanta dan syarat awal

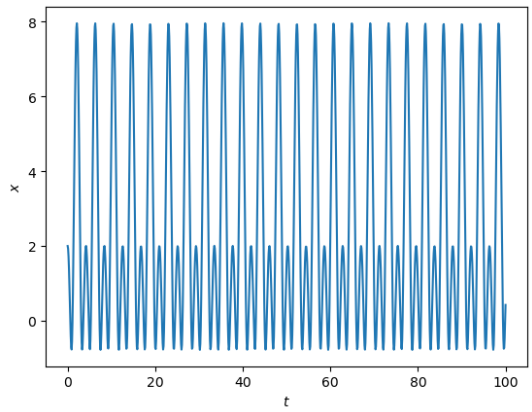
Selanjutnya data yang diperoleh untuk membuat animasi. Karena animasi memerlukan data posisi (x, y) sehingga perlu adanya transformasi koordinat. Berikut transformasi koordinatnya

$$x_1 = (l + x)\sin\theta$$

$$y_1 = -(l + x)\cos\theta$$



Gambar 5. Grafik θ vs t



Gambar 6. Grafik x vs t

Data x dan θ diperoleh dari transpose matrik dalam variabel ans orde [0] dan [2]. Dari data x_1 dan y_1 diperoleh dengan substitusi x dan θ ke transformasi koordinat. Animasi disimpan dalam bentuk gif.

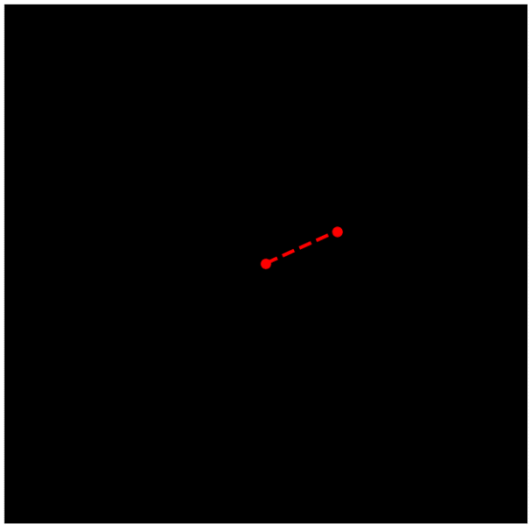
```
def animate(i):
    ln1.set_data([0, x1[i]], [0, y1[i]])

fig, ax = plt.subplots(1,1, figsize=(8,8))
ax.set_facecolor('k')
ax.get_xaxis().set_ticks([])    # enable this to hide x axis ticks
ax.get_yaxis().set_ticks([])    # enable this to hide y axis ticks
ln1, = plt.plot([], [], 'ro--', lw=3, markersize=8)
ax.set_ylim(-15,15)
ax.set_xlim(-15,15)
ani = animation.FuncAnimation(fig, animate, frames=1000, interval=50)
ani.save('spring_pendulum.gif',writer='pillow',fps=25)
```

Gambar 7. Coding pembuatan animasi

Hasil perbandingan animasi gif yang dibuat dengan Python dengan animasi dari web (Neumann, 2021) adalah hampir sama dengan pendekatan Euler-Lagrange.

p-ISSN:2527-5321
e-ISSN:2527-5941



Gambar 8. Visualisasi sederhana dari pendulum pegas (titik merah tengah titik gantung, titik merah sebelah kanan massa *m*)

SymPy sudah ada versi yang terbaru (saat artikel ini dibuat) yaitu versi 1.12. akan tetapi pada saat menjalankan perintah untuk menyelesaikan persamaan terjadi error (proses tidak berhenti atau running terus-menerus) sehingga dipilihlah SymPy versi 1.6.2.

SIMPULAN DAN SARAN

SymPy versi 1.6.2 sudah mumpuni untuk penyelesaian tugas analisis persamaan secara simbolis walaupun perlu penyeseuaian. Kemungkinannya kemampuan Library SymPy ini juga cukup mampu digunakan untuk sistem yang kompleks.

Karena library SymPy untuk sistem di atas dapat menyelesaikan dengan baik, kami sarankan untuk penelitian selanjutnya adalah dicoba untuk sistem yang lebih kompleks dan untuk sistem kuantum untuk menghindari analisis manual atau tulis tangan.

DAFTAR PUSTAKA

Abohamer, M. K., Awrejcewicz, J., Starosta, R., Amer, T. S., & Bek, M. A. (2021). Influence of the motion of a spring pendulum on energy-harvesting devices. *Applied Sciences (Switzerland)*, *11*(18). https://doi.org/10.3390/app11188658

Baleanu, D., Asad, J. H., & Jajarmi, A. (2018). The Fractional Model of Spring Pendulum: New Features within Different Kernels. *Proceedings Of The Romanian Academy, Series A*, *19*(3), 447–454.

Boas, M. L. (1983). *Mathematical Methods in The Physical Sciences* (2nd ed.). John Wiley & Sons.

Broucke, R., & Baxa, P. A. (1973). Periodic Solutions of A Spring-Pendulum System*. *Conference on Celestial Mechanics*, 261–267.

De Sousa, M. C., Marcus, F. A., Caldas, I. L., & Viana, R. L. (2017). Energy Distribution in Spring Pendulums. In *Arxiv*.

Morin, D. J. (2008). *Introduction to Classical Mechanics : With Problems and Solutions*. Cambridge University Press.

Neumann, E. (2021, June 25). *2D Spring*. Https://Www.Myphysicslab.Com/Springs/2d-Spring-En.Html.

Solve an Equation Algebraically. (2023). Https://Docs.Sympy.Org/Latest/Guides/Solving/Solve-Equation-Algebraically.Html.

Weber, H. J., & Arfken, G. B. (2004). *Essential Mathematical Methods for Physicists* (6th ed.). Academic Press.

Wells, D. A. (1967). Schaum's Outline: Lagrangian Dynamics. In *Schaum's Outline Series*. McGraw-Hill.