PRAKTIKUM FISIKA KOMPUTASI

MODUL-8-ANIMASI DOUBLE PENDULUM

Disusun Oleh:

Dewi Rahawati(1227030010)

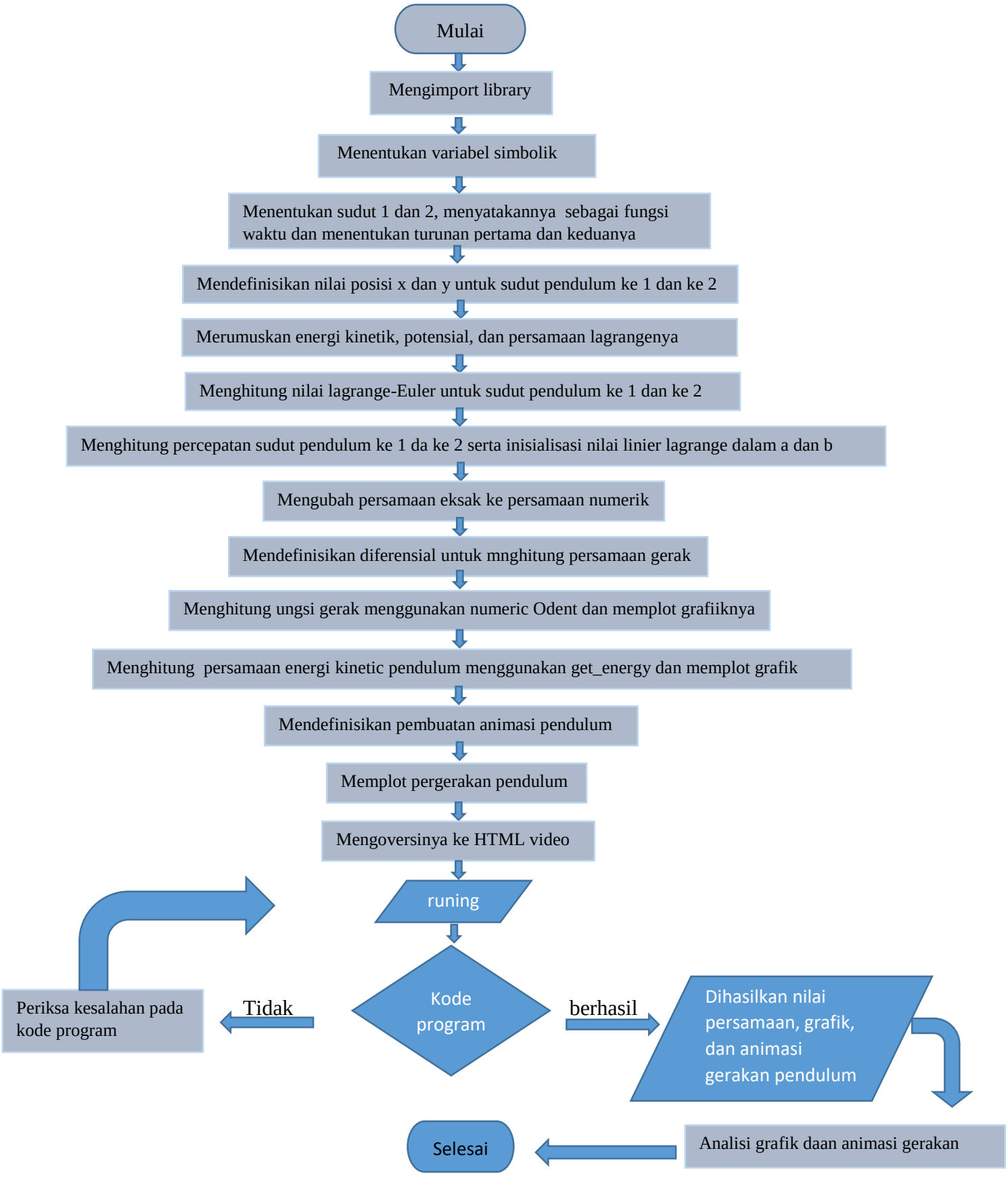
Fisika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sunan Gunung Djati Bandung

2024

1. *Flow Chart*


Mulai
Mengimport library
Menentukan variabel simbolik
Menentukan sudut 1 dan 2, menyatakannya sebagai fungsi waktu dan menentukan turunan pertama dan keduanya
Mendefinisikan nilai posisi x dan y untuk sudut pendulum ke 1 dan ke 2
Merumuskan energi kinetik, potensial, dan persamaan lagrangenya
Menghitung nilai lagrange-Euler untuk sudut pendulum ke 1 dan ke 2
Menghitung percepatan sudut pendulum ke 1 da ke 2 serta inisialisasi nilai linier lagrange dalam a dan b
Mengubah persamaan eksak ke persamaan numerik
Mendefinisikan diferensial untuk mnghitung persamaan gerak
Menghitung ungsi gerak menggunakan numeric Odent dan memplot grafiiknya
Menghitung persamaan energi kinetic pendulum menggunakan get_energy dan memplot grafik
Mendefinisikan pembuatan animasi pendulum
Memplot pergerakan pendulum
Mengoversinya ke HTML video
runing
Kode program
Tidak
Periksa kesalahan pada kode program
berhasil
Dihasilkan nilai persamaan, grafik, dan animasi gerakan pendulum
Analisi grafik daan animasi gerakan
Selesai

2. Kode program

```
import numpy as np
import sympy as smp
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from matplotlib import animation
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import PillowWriter
from IPython.display import HTML
```

Kode program diatas digunakan untuk menginput library yang digunakan pada praktikum. Library umum yang digunakannya yaitu berupa library numpy, sympy, odeint, matplotlib, dan Ipython display. Library nump digunakan untuk perhitungan secara numerik. Library sympy digunakan untuk perhitungan simbolik untuk turunan yang akan digunakan. Library matplotlib yang diigunakan untuk memplot grafik dan meganimasikan video engan HTML.

```
Menentukan variable yang diperlukan ntuk sympy
t, m, g, L1, L2, w, C, alph, beta = smp.symbols(r't m g L_1 L_2 \omega C \alpha \beta')
```

Kode program diatas digunakan untuk menginisiasi variable untuk perhitungan sympy.

```
#Menentukan tetha 1 dan theta 2dan menyatakn fungsi waktjuga definisi turunn pertama dan kedua
the1, the2, = smp.symbols(r' \theta_1, \theta_2', cls=smp.Function)

the1 = the1(t)
the1_d = smp.diff(the1, t)
the1_dd = smp.diff(the1_d, t)

the2 = the2(t)
the2_d = smp.diff(the2, t)
the2_dd = smp.diff(smp.diff(the2, t), t)
```

Kode program diatas digunakan untuk menentukan nilai sudut pendulum pertama dan kedua serta perhitungan turunan kesatu dan kedua untuk setiap sudutnya.

```
#Mendeklarasikan nilai x1(theta1), y1(theta1), dan x2(theta1, theta2), y2(theta1,thheta2)
x1,y1, x2, y2 = smp.symbols('x_1, y_1, x_2, y_2', cls=smp.Function)
x1= x1(t, the1)
y1= y1(t, the1)
x2= x2(t, the1, the2)
y2= y2(t, the1, the2)

#Masukkan ke dalam bentuk fungsionalspesifik dari x1,1,x2,y2
x1 = smp.cos(w*t)+L1*smp.sin(the1)
```

```
y1 = -L1*smp.cos(the1)
x2 = smp.cos(w*t)+L1*smp.sin(the1) +L2*smp.sin(the2)
y2 = -L1*smp.cos(the1) -L2*smp.cos(the2)
```

Kode program diatas digunakan untuk menentukan posisi sudut pendulum kesatu dan kedua pada saat waktu tertentu dengan posisi arah x dan y yatu, x1,y1,x2,y2 serta menghitung nilai posisinya.

```
#definisi fungsi numerik dari vx1, vy1, vx2,vy2
smp.diff(x1,t)

vx1_f = smp.lambdify((t,w,L1,L2,the1,the2,the1_d,the2_d), smp.diff(x1, t))
vx2_f = smp.lambdify((t,w,L1,L2,the1,the2,the1_d,the2_d), smp.diff(x2, t))
vy1_f = smp.lambdify((t,w,L1,L2,the1,the2,the1_d,the2_d), smp.diff(y1, t))
vy2_f = smp.lambdify((t,w,L1,L2,the1,the2,the1_d,the2_d), smp.diff(y2, t))
```

Kode program tersebut digunakan untuk menentukan nilai kecepatan pendulum pada posisi x1,y1,x2,y2 yaitu pada saat sudut kesatu dan sudut kedua.

```
#rumus lagrange
T = 1/2 * (smp.diff(x1, t)**2 + smp.diff(y1, t)**2) + \
    1/2 * m * (smp.diff(x2, t)**2 + + smp.diff(y2, t)**2)
V = g*y1 + m*g*y2
L = T-V
```

Kode program diatas merupakan kode program untuk meentukan nilai energi kinetik sebagai T dan energi potensial sebagai V serta nilai lagrangenya sebagai pengurangan dari energi kinetik dan potensialnya.

```
#Persamaan Lagrange-Euleur untuk theta1
LE1 = smp.diff(L, the1) - smp.diff(L, the1_d, t)
LE1 = LE1.simplify()

#Persamaan Lagrange-theta untuk theta2
LE2 = smp.diff(L, the2) - smp.diff(L, the2_d, t)
LE2 = LE2.simplify()

LE1
LE2
sols =  smp.solve([LE1, LE2], (the1_dd, the2_dd),
                simplify=False, rational=False)
sols[the1_dd] #d^2 / dt^2 theta_1
```

Kode program tersebut merupakan persamaan untuk mencari nilai lagrange sudut kesatu dan kedua dengan menggunakan persamaan lagrange-Euler serta kode program untuk menentukan percepatan sudutnya.

```
a = LE1.subs([(smp.sin(the1 - the2), the1 - the2),
    (smp.cos(the1 - the2), 1),
    (smp.cos(the1), 1),
    (smp.sin(the1), the1),
    (the1, C * smp.cos(w * t)),
    (the2, C * alph * smp.cos(w * t)),
    (m, 1),
    (L2, L1),
]).doit().series(C, 0, 2).removeO().simplify()

b = LE2.subs([(smp.sin(the1 - the2), the1 - the2),
    (smp.cos(the1 - the2), 1),
    (smp.cos(the1), 1),
    (smp.sin(the1), the1),
    (smp.sin(the2), the2),
    (the1, C * smp.cos(w * t)),
    (the2, C * alph * smp.cos(w * t)),
    (m, 1),
    (L2, L1),
]).doit().series(C, 0, 2).removeO().simplify()

yeet = smp.solve([a.args[1], b.args[2]], (w, alph))

yeet[2][0]
yeet[0][0]

smp.limit(yeet[1][0].subs(C, beta/L1).simplify(), beta, smp.oo)
```

Kode program tersebut merupakan kode program untuk menentukan bentuk linear dari lagrange pada sudut pertama dan kedua

```
dz1dt_f = smp.lambdify((t, m, g, w, L1, L2, the1, the2, the1_d, the2_d), sols[the1_dd])
dthe1dt_f = smp.lambdify(the1_d, the1_d)

dz2dt_f = smp.lambdify((t, m, g, w, L1, L2, the1, the2, the1_d, the2_d), sols[the2_dd])
dthe2dt_f = smp.lambdify(the2_d, the2_d)
```

Kode program tersebut adalah untuk mengubah persamaan eksak menjadi persamaan numeric

```
def dSdt(S, t):
  the1, z1, the2, z2 = S
  return [
      dthe1dt_f(z1),
      dz1dt_f(t, m, g, w, L1, L2, the1, the2, z1, z2),
      dthe2dt_f(z2),
```

```
    dz2dt_f(t, m, g, w, L1, L2, the1, the2, z1, z2),
]
```

Kode program tersebut digunakan untuk menghitung persamaan geraknya.

```
t = np.linspace(0, 20, 1000)
g = 9.81
m = 1
L1 = 20
L2 = 20
w = np.sqrt(g/L1)
ans = odeint(dSdt, y0=[0, 0, 0, 0], t=t)

plt.plot(ans.T[0])
```

Kode program tersebut digunakan untuk menentukan menghitung persamaan gerak dengan menggunakan numerik odent dan diplot grafiknya.

```
def get_energy(w):
    t = np.linspace(0, 100, 2000)
    ans = odeint(dSdt, y0=[0.1, 0.1, 0, 0], t=t)
    vx1 = vx1_f(t,w,L1,L2,ans.T[0],ans.T[2],ans.T[1],ans.T[3])
    vx2 = vx2_f(t,w,L1,L2,ans.T[0],ans.T[2],ans.T[1],ans.T[3])
    vy1 = vy1_f(t,w,L1,L2,ans.T[0],ans.T[2],ans.T[1],ans.T[3])
    vy2 = vy2_f(t,w,L1,L2,ans.T[0],ans.T[2],ans.T[1],ans.T[3])
    E = 1/2 * np.mean(vx1**2 + vx2**2 + vy1**2 + vy2**2)
    return E


ws = np.linspace(0.4, 1.3, 100)
Es = np.vectorize(get_energy)(ws)

plt.plot(ws, Es)
plt.axvline(1.84775 * np.sqrt(g / L1), c='k', ls='--')
plt.axvline(0.76536 * np.sqrt(g / L1), c='k', ls='--')

plt.grid()
```

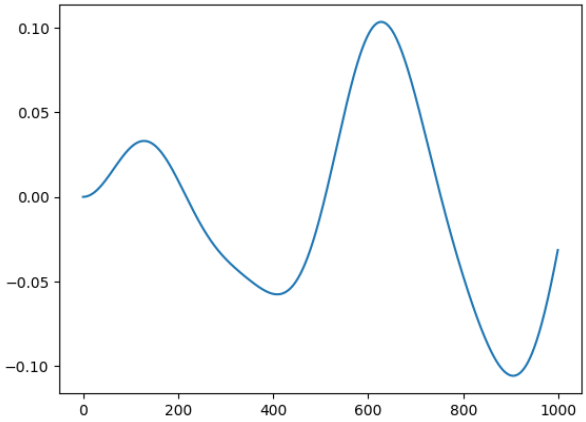
Kode program tersebut digunakan untuk mencari nilai dari persamaan energi kinetik dengan menggunakan get_energy dan memplot grafiknya.

```
t = np.linspace(0, 200, 20000)
g = 9.81
m = 1
```

```
L1 = 20
L2 = 20
w = ws[ws>1][np.argmax(Es[ws>1])]
ans = odeint(dSdt, y0=[0.1, 0.1, 0, 0], t=t)

def get_x0y0x1y1x2y2(t, the1, the2, L1, L2):
    return (np.cos(w*t),
            0*t,
            np.cos(w*t) + L1*np.sin(the1),
            -L1*np.cos(the1),
            np.cos(w*t) + L1*np.sin(the1) + L2*np.sin(the2),
            -L1*np.cos(the1) - L2*np.cos(the2)
      )
x0, y0, x1, y1, x2, y2 = get_x0y0x1y1x2y2(t, ans.T[0], ans.T[2], L1, L2)

def animate(i):
    ln1.set_data([x0[::10][i], x1[::10][i], x2[::10][i]], [y0[::10][i],
y1[::10][i], y2[::10][i]])
    trail1 = 50
    trail2 = 50
    ln2.set_data(x1[::10][i:max(1,i-trail1):-1], y1[::10][i:max(1,i-
trail1):-1]) # Use the built-in function max()
    ln3.set_data(x2[::10][i:max(1,i-trail2):-1], y2[::10][i:max(1,i-
trail2):-1]) # Use the built-in function max()

fig, ax = plt.subplots(1, 1, figsize=(8, 8))
ax.set_facecolor('k')

ax.get_xaxis().set_ticks([])
ax.get_yaxis().set_ticks([])

ln1, = ax.plot([], [], 'ro--', lw=3, markersize=8)
ln2, = ax.plot([], [], 'ro-', markersize=8, alpha=0.05, color='cyan')
ln3, = ax.plot([], [], 'ro-', markersize=8, alpha=0.05, color='cyan')

ax.set_ylim(-44, 44)
ax.set_xlim(-44, 44)

ani = animation.FuncAnimation(fig, animate, frames=2000, interval=50)
HTML(ani.to_html5_video())
```

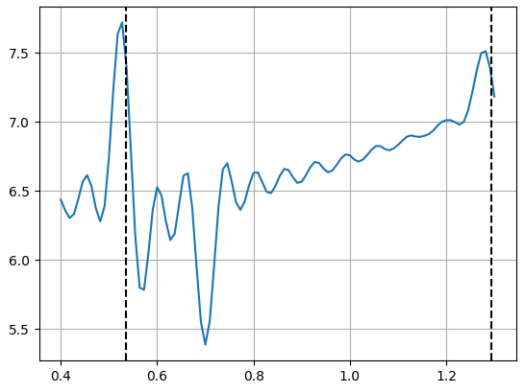
Kode program tersebut berisi defisini yang dibutuhkan untuk membuat pendulum, memplot pendulumnya serta menampilkan pendulumny berupa video.

3. Analisis Grafik
   a. Grafik gerak sistem pendulum terhadap waktu



Grafik tersebut merupakan grafik hubungan antara gerak pendulum pada persamaan lagrange yang mencakup posisi sudut ke 1 dan ke 2 serta turunan pertama dan kedua dari kedua sudut pendulum tersebut terhadap waku dari 0 sekon sampai 20 sekon dengan interval 1000. Pada grafik dapat dilihat bahwa pergerakan sistem pendulum berubah seiring berjalannya waktu. Semakin lama waktu yang diperlukan untuk pendulum bergerak osilasi maka nilai amplitudonya akan semakin besar. Hal tersebut berarti gerak kedua pendulum pada masing-masing sudut tidak bersifat harmonik atau periodik. Ketidaharmonikan gerak pendulum tersebut bisa terjadi karena pergerakan pendulum pertama atau pada sudut kesatu dipengaruhi oleh pergerakan pada sudut kedua atau pendulum kedua. Begitupun sebaliknya, pergerakan pendulum pada sudut kedua dipengaruhi oleh pergerakan atau osilasi pendulum sudut pertama.

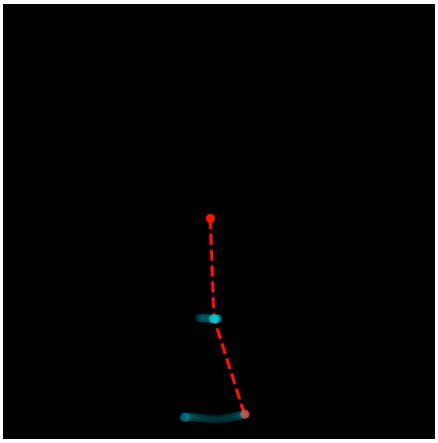
   b. Grafik energi terhadap frekuensi



Grafik tersebut merupakan grafik hubungan antara energi gerak pendulum terhadap frekuensi. Ketika pendulum bergerak maka pendulum tersebut akan memiliki energi. Pendulum juga memiliki frekuensi akibat dari pergerakannya terhadap waktu. Jika dilihat dari grafik maka dapat dilihat bahwa nilai besarnya energi pada pendulum terus berubah-ubah sesuai dengan nilai frekuensimnya. Hubungan keduanya menghasilkan

pola gelombang sinusoidal yang terdiri atas puncak dan lembah. Pada grafik juga dapat dilihat bahwa energi yang besar atau puncak tertingginya ada di frekuensi yang mendekati 0,6 atau sekitar 0,53 nilai energinya yaitu sekitar 8.0.

c. Animasi pendulum



Pada animasi gerak pendulum dapat dilihat bahwa pergerakan antara pendulum sudut kesatu dan kedua itu saling mempengaruhi. pendulum pertama atau pada sudut kesatu dipengaruhi oleh pergerakan pada sudut kedua atau pendulum kedua. Begitupun sebaliknya, pergerakan pendulum pada sudut kedua dipengaruhi oleh pergerakan atau osilasi pendulum sudut pertama.