# Laporan Tugas Besar
# Simulasi Gerak Pendulum Menggunakan Metode Euler Berbasis Pemograman Bahasa C


Disusun oleh:

Kelompok 11 Kelas TF-47-03

Anggota:

Umar Zaki Gunawan 101042330035

Violola Aprilia 101042330085

Steven Panjaitan 101042330062

FAKULTAS TEKNIK ELEKTRO

TELKOM UNIVERSITY

BANDUNG

2023

# Latar Belakang

Pada kehidupan kita pasti pernah meilihat peristiwa gerak pendulum atau gerak bandul. Contohnya suatu permainan yang pernah kita mainkan saat kecil dahulu yaitu lato-lato atau juga ayunan. Gerakan bolak-balik yang dilakukan oleh benda yang kurang lebih berat, disebut pendulum, yang digantung dengan tali atau batang ringan di satu ujung sisi dan dibiarkan berosilasi di ujung sisi lain, dengan cara ini sistem tersebut menggambarkan busur bolak-balik. Fenomena ini memiliki peran penting dalam memahami konsep-konsep dasar dalam fisika, seperti energi kinetik dan potensial, serta prinsip-prinsip mekanika Newton.

Untuk memberikan gambaran bagaimana persitiwa gerak pendulum bekerja, pada laporan kali ini kami menganalisis menggunakan metode numerik yaitu model Euler Kromus.Metode Euler memungkinkan untuk memecahkan persamaan diferensial yang menggambarkan gerak bandul secara numerik. Metode ini memungkinkan pendekatan perhitungan iteratif dengan menggunakan langkah-langkah waktu kecil untuk memperkirakan posisi dan kecepatan bandul pada setiap titik waktu. Dengan demikian, kelompok kami menerapkan Metode Euler ke dalam sebuah pemrograman. Dengan ini dapat kita harapkan untuk memvisualisasikan dan menganalisis perubahan posisi, kecepatan, dan energi yang terjadi pada Gerak Pendulum.

# Metode

Metode Euler adalah salah satu metode numerik yang digunakan untuk menyelesaikan persamaan diferensial biasa (ODE) secara aproksimatif. Ditemukan oleh matematikawan Swiss Leonhard Euler, metode ini digunakan untuk mengaproksimasi solusi dari persamaan diferensial dengan membagi interval waktu tertentu menjadi langkah-langkah kecil.Metode numerik yang digunakan yaitu Metode Euler.

Terdapat 2 jenis metode euler yaitu metode euler implisit dan metode euler eksplisit. Metode euler eksplisit atau metode Euler Kromer lebih akurat dibandingkan implisit. Metode ini hanya membutuhkan satu titik nilai yang diketahui sehingga dapat disebut metode titik tunggal. Persamaan umum untuk metode euler eksplisit adalah:

$$y(t+\Delta t) = y(t) + \Delta t f(t) \quad (1)$$

dimana $\Delta t$ merupakan selang waktu. Semakin kecil nilai $\Delta t$ maka nilai akan semakin akurat. Persamaan gerak dari pendulum teredam disubstitusikan kedalam persamaan metode euler tersebut, menjadi

$$\omega_{n+1} = \omega_n + (-g/l) * \theta * dt \quad (2)$$

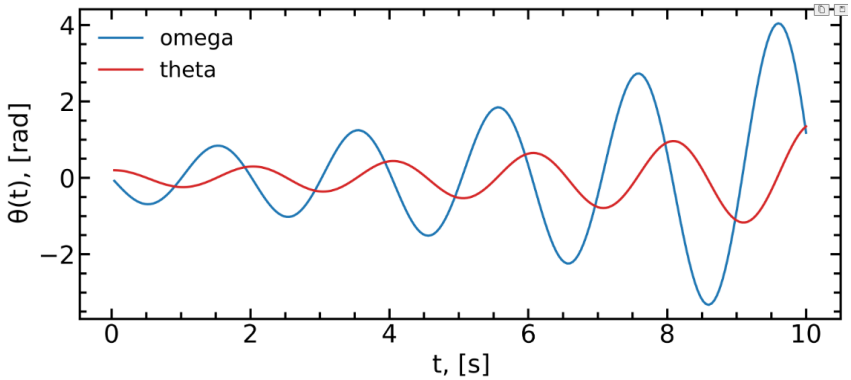
$$\theta_{n+1} = \theta_n + \omega_n * dt \quad (3)$$

Persamaan (2) dan persamaan (3) diatas akan digunakan dalam program kami menggunakan perulangan di mana kita menetapkan jumlah perulangan dengan variabel npoints sebanyak 250 kali. Namun sebelum masuk ke operasi utama di atas pertama-tama kami menetapkan variabel omega, sudut, dan juga waktu menjadi sebuah array. Setelahnya array tersebut ditetapkan ukuran sebanyak npoints lalu seluruh isinya dikosongkan terlebih dahulu dengan menggunakan perulangan sebanyak npoints. Lalu untuk output dari program tersebut akan disimpan didalam sebuah file eksternal dimana fungsi dari file tersebut untuk membuat plotting grafik.

# Hasil dan Pembahasan

Simulasi gerak pendulum pada laporan ini bertujuan untuk mensimulasikan dan mempelajari gerak bandul sederhana dengan menggunakan pemrograman bahasa C lalu membandingkan metode euler sederhana dan euler kromer. Berikut ini merupakan hasil ploting grafik menggunakan bahasa python jupyter.

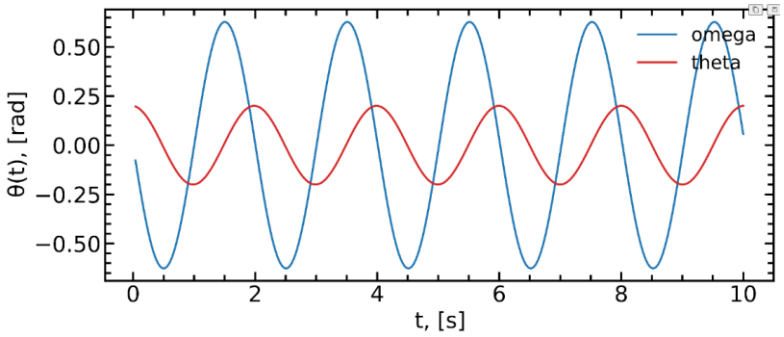
## Metode Euler Normal




Dapat dilihat dari hasil grafik pada gambar diatas bahwa baik itu omega dan juga theta memiliki pola yang berulang sampai tak hingga karena dalam simulasi ini tidak ditambahkan faktor hambatan udara yang menyebabkan pergerakan bandul tersebut berhenti. Perbdaan dari omega dan theta sendiri terletak pada fase yang berbeda.

Pada grafik menggunakan metode euler sendiri jika disadari terdapat adanya keanehan dimana semakin bertambahnya waktu grafik menunjukan adanya perbesaran. Hal tersebut dikatakan tidak mungkin karena seharusnya energi yang ada tidak ada penambahan. Atau dalam hal ini karena tidak adanya hambatan maka seharusnya grafik menunjukan besar amplitudo gelombang yang tetap.

## Metode Euler Kromer




Dapat dilihat dari gambar diatas grafik sudah menunjukan gambar yang seharusnya di mana ampilitudo tetap dari awal hingga akhir.

# Kesimpulan

Metode Euler dan Metode Euler-Kromer adalah dua pendekatan yang serupa dalam menyelesaikan persamaan diferensial numerik, tetapi memiliki perbedaan dalam cara mereka memperbarui variabel tergantung pada waktu. Dalam hal ini metode euler kromer lebih dibutuhkan karena penggambaran simulasi gerak pendulum menggunakan metode euler kromer lebih akurat dibandingkan dengan metode Euler Normal.

Perbedaan utama :

- **Urutan Pembaruan Variabel**: Metode Euler menghitung nilai variabel terlebih dahulu, lalu memperbarui waktu. Sementara itu, Metode Euler-Kromer memperbarui nilai variabel terlebih dahulu, lalu baru memperbarui waktu.
- **Akibatnya**: Hal ini dapat mengakibatkan perbedaan kecil dalam akurasi metode, terutama ketika menangani persamaan yang terkait erat dengan energi atau sistem yang konservatif. Metode Euler-Kromer cenderung lebih stabil karena memperbarui waktu setelah memperbarui variabel lainnya

**Lampiran Kode**

Bandul.c (Metode Euler Normal)

```c
#define _CRT_SECURE_NO_WARNINGS
#include <math.h>
#include <stdio.h>
void main() {
    FILE *fptr;
    FILE *fptr2;
    int i,npoints;
    double length,g,dt,omega[250],theta[250],time[250];
    /* Two output files - one measuring omega, one measuring theta */
    fptr=fopen("pendout.dat","w");
    fptr2=fopen("pendoutb.dat","w");
    /* Preset the parameters from the formula */
    length=1.0; /* Preset length of pendulum (L) */
    g=9.8; /* Preset acceleration of gravity (m/s^2) */
    npoints=250; /* Preset number of points in loop */
    dt=0.04; /* Preset time interval (s) */
    /* Clear storage arrays to zero */
    for(i=0;i<npoints;i++) {
        omega[i]=0.0;
        theta[i]=0.0;
        time[i]=0.0;
    }
    /* Preset initial theta and omega values */
    theta[0]=0.2;
    omega[0]=0.0;
    /* Euler method */
    /* ωn+1 = ωn + (-g / L )* θ*dt */
    /* and */
    /* θn+1 = θn + ωn*dt */
    for(i=0;i<npoints;i++) {
        omega[i+1]=omega[i]-(g/length)*theta[i]*dt;
        theta[i+1]=theta[i]+omega[i]*dt;
        time[i+1]=time[i]+dt;
        fprintf(fptr,"%lf\t%lf\n",time[i+1],theta[i+1]);
        fprintf(fptr2,"%lf\t%lf\n",time[i+1],omega[i+1]);
    }
    fclose(fptr);
    fclose(fptr2);
}
```

Bandul2.c (Metode Euler Kromer)

```c
/* pendme2.c */
/*
Euler-Cromer method
 */
#define _CRT_SECURE_NO_WARNINGS
#include <math.h>
#include <stdio.h>
void main() {
    FILE *fptr;
    FILE *fptr2;
    int i,npoints;
    double length,g,dt,omega[250],theta[250],time[250];
    /* Two output files - one measuring omega, one measuring theta
    fptr=fopen("pendout2.dat","w");
    fptr2=fopen("pendout2b.dat","w");
    /* Preset the parameters from the formula */
    length=1.0; /* Preset length of pendulum (L) */
    g=9.8; /* Preset acceleration of gravity (m/s2) */

    npoints=250; /* Preset number of points in loop */
    dt=0.04; /* Preset time interval (s) */
    /* Clear storage arrays to zero */
    for(i=0;i<npoints;i++) {
        omega[i]=0.0;
        theta[i]=0.0;
        time[i]=0.0;
    }
    /* Preset initial theta and omega values */
    theta[0]=0.2;
    omega[0]=0.0;
    /* Euler-Cromer method */
    /* ωn+1 = ωn + (-g / L )* θ*dt */
    /* and */
    /* θn+1 = θn + ωn+1*dt*/
    for(i=0;i<npoints;i++) {
        omega[i+1]=omega[i]-(g/length)*theta[i]*dt;
        theta[i+1]=theta[i]+omega[i+1]*dt;
        time[i+1]=time[i]+dt;
        printf("%Lf %Lf\n", time[i+1], theta[i+1]);
        printf("%Lf %Lf\n", time[i+1], omega[i+1]);
        fprintf(fptr,"%Lf\t%Lf\n",time[i+1],theta[i+1]);
        fprintf(fptr2,"%Lf\t%Lf\n",time[i+1],omega[i+1]);
    }
    fclose(fptr);
    fclose(fptr2);
    }
```

Plot-dos.ipynb (Ploting metode euler normal)

```
import matplotlib.pyplot as plt
plt.style.use('style/sci.mplstyle')
import numpy as np
figsize = (12, 5)
dpi = 600

x , y = np.loadtxt('pendout2.dat', unpack=True)
a, b = np.loadtxt('pendout2b.dat', unpack=True)

# Create figure object
plt.figure(figsize=figsize, dpi=dpi)
# Plot the DOS, in which the Fermi energy shifts to zero
plt.plot(a,b, label='omega')
plt.plot(x,y, label='theta')
plt.xlabel(r"$t$, [s]")
plt.ylabel(r"$\theta(t)$, [rad]")
plt.legend()
plt.show()
plt.savefig('plot-dos.jpg')
```

Plot-dos2.ipynb (Plotting metode euler kromer)

```
import matplotlib.pyplot as plt
plt.style.use('style/sci.mplstyle')
import numpy as np
figsize = (12, 5)
dpi = 600

x , y = np.loadtxt('pendout.dat', unpack=True)
a, b = np.loadtxt('pendoutb.dat', unpack=True)

# Create figure object
plt.figure(figsize=figsize, dpi=dpi)
plt.plot(a,b, label='omega')
plt.plot(x,y, label='theta')
plt.xlabel(r"$t$, [s]")
plt.ylabel(r"$\theta(t)$, [rad]")
plt.legend()
plt.show()
plt.savefig('plot-dos.pdf')
```