

Diktat Praktikum

KOMPUTASI DAN SIMULASI FISIKA

UND 2025 14004/Semester 4

Oleh:

Dr. Budi Adiperdana, M.Si.

Ferry Faizal, Ph.D.

Laboratorium Komputasi

Departemen Fisika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

Jatinangor, Februari 2026

MODUL PRAKTIKUM KOMPUTASI DAN SIMULASI FISIKA

Copyright © 2026 – UND202514004 / 3rd Semester, Laboratorium of Computation.

Diktat ini merupakan panduan bagi mahasiswa S-1 *Fisika Unpad* dalam memahami mata kuliah komputasi dan simulasi fisika secara praktek.

- *Departemen Fisika, Universitas Padjadjaran*

Modul-1: Model Persamaan Diferensial

Modul-2: Stabilitas Numerik dan Resolusi Diskritisasi

Modul-3: Skema Waktu: Metode Eksplisit dan Implisit

Modul-4: Gerak Partikel, Syarat Batas dan Gerak Realistik (Air drag, Air density)

Modul-5: Gerak Osilasi (Bandul dan Pegas)

Modul-6: Bilangan Acak dan Fenomena Stokastik (Integral di kelas, Random Walk)

Modul-7: Persamaan Medan: Laplace dan Poisson

Modul-8: Persamaan Evolusi Kontinu: Difusi

Modul-9: Persamaan Evolusi Kontinu: Gelombang

Projek Simulasi

MODUL – 3: Gerak Partikel, Syarat Batas dan Gerak Realistik (Air Drag, Air Density)

1. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Merekonstruksi kasus fisis gerak jatuh bebas (dengan/tanpa hambatan udara) dan gerak peluru ke dalam model matematika.
2. Mengungkapkan model fisis ke dalam algoritma numerik (Metode Euler).
3. Menampilkan solusi dalam bentuk grafik lintasan dan animasi dinamis.
4. Merekonstruksi kasus fisis gerak jatuh bebas dan gerak peluru ke dalam model matematika menggunakan *Dirichlet – Neumann Boundary Condition*.

2. Teori

1. Gerak Jatuh Bebas dengan Hambatan Udara

Pada GJB murni, hanya gaya gravitasi yang bekerja. Namun, untuk rekonstruksi yang lebih realistis, kita memasukkan gaya hambat udara (F_d) yang sebanding dengan kecepatan:

$$m \frac{dv}{dt} = mg - bv \rightarrow \frac{dv}{dt} = g - \frac{b}{m} v$$

Di mana b adalah koefisien hambat udara.

2. Gerak Peluru (*Projectil Motion*)

Gerak peluru adalah kombinasi gerak horizontal (GLB) dan vertikal (GLBB). Jika hambatan udara diabaikan, persamaannya adalah:

$$\text{Sumbu } x : \frac{d^2x}{dt^2} = 0$$

$$\text{Sumbu } y : \frac{d^2y}{dt^2} = -g$$

Untuk pendekatan numerik, kita memecahnya menjadi sistem empat PDB orde satu:

$$\frac{dx}{dt} = vx, \frac{dv_x}{dt} = 0$$

$$\frac{dy}{dt} = vy, \frac{dv_y}{dt} = -g$$

3. Implementasi Kode (Python)

Berikut adalah simulasi gerak peluru menggunakan metode Euler, lengkap dengan visualisasi animasi

Kode-2.2

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

```

from IPython.display import HTML
# --- 1. Parameter Fisis ---
g = 9.81
v0 = 20.0
theta = 45.0
dt = 0.05
e = 0.6          # Koefisien restitusi (Neumann: seberapa besar vy dibalik)
v_stop = 1.2     # Threshold berhenti (Absorb condition)
vx = v0 * np.cos(np.radians(theta))
vy = v0 * np.sin(np.radians(theta))
# --- 2. Inisialisasi State [x, y, vx, vy] ---
# Initial Condition: semua nilai ditetapkan di t=0
state = np.array([0.0, 0.0, vx, vy])
path_x, path_y = [], []
# --- 3. Loop Numerik: Metode Euler ---
running = True
while running:
    path_x.append(state[0])
    path_y.append(state[1])
    # Update posisi & kecepatan (IVP: maju dalam waktu)
    state[0] += state[2] * dt
    state[1] += state[3] * dt
    state[3] -= g * dt
    if state[1] < 0:
        # [DIRICHLET b.c.] → nilai posisi ditetapkan: y = 0
        state[1] = 0.0
        # [NEUMANN b.c.] → turunan (vy) dibalik & diredam:  $\partial y / \partial n = f$ 
        state[3] = -state[3] * e
        state[2] = state[2] * e    # gesekan horizontal
        # [ABSORB condition] → berhenti jika energi habis
        speed = np.sqrt(state[2]**2 + state[3]**2)
        if speed < v_stop:
            running = False
# --- 4. Animasi ---
fig, ax = plt.subplots(figsize=(8, 3))
ax.set_xlim(0, max(path_x) + 5)
ax.set_ylim(0, max(path_y) + 5)
ax.set_aspect('equal')
ax.axhline(0, color='brown', linewidth=2, label='Tanah')
line, = ax.plot([], [], 'o', color='red', markersize=8, label='Peluru')
trail, = ax.plot([], [], '--', color='gray', alpha=0.5, label='Lintasan')
ax.legend()

def update(frame):
    trail.set_data(path_x[:frame], path_y[:frame])
    line.set_data([path_x[frame]], [path_y[frame]])
    return line, trail
ani = FuncAnimation(fig, update, frames=len(path_x), interval=30, blit=True)

```

```
plt.xlabel("Jarak (m)")
plt.ylabel("Tinggi (m)")
plt.grid(True)
plt.close()
HTML(ani.to_jshtml())
```

4. Syarat Batas (*Boundary Condition*)

Dalam simulasi gerak partikel, sistem dibatasi oleh suatu domain komputasi. Ketika partikel mencapai batas domain tersebut, kita perlu mendefinisikan **syarat batas** (*boundary conditions*, b.c.) yang menentukan bagaimana partikel berinteraksi dengan batas. Terdapat dua jenis syarat batas yang paling umum digunakan:

4.1. Syarat Batas *Dirichlet*

Syarat batas **Dirichlet** mendefinisikan **nilai fungsi secara langsung** pada batas domain:

$$y = f$$

Dalam konteks gerak partikel, ini berarti kita **mengunci nilai posisi** partikel ketika mencapai batas. Beberapa implementasinya:

Kondisi Fisis	Implementasi Kode	Keterangan
Berhenti (Absorb/Stop)	if $y[i] < 0$: $y[i]=0$; $vy[i]=0$	Partikel berhenti tepat di batas
Memantul (Reflect)	if $y[i] < 0$: $y[i]=-y[i]$; $vy[i]=-vy[i]$	Posisi dicerminkan, kecepatan dibalik
Memantul Teredam	if $y[i] < 0$: $y[i]=-y[i]$; $vy[i]=-c*vy[i]$	$c<1$ teredam, $c>1$ dipercepat
Wrap Around (Periodic)	if $y[i] < 0$: $y[i]=y_0$; $vy[i]=vy[i]$	Partikel muncul kembali di sisi lain

Contoh kode untuk kasus gerak jatuh bebas, bola memantul dengan redaman di lantai ($y = 0$):

Kode-2.2

```
# Syarat Batas Dirichlet -- Memantul Teredam
c = 0.8 #koefisien resitusi (0 < c <= 1)
```

```
y[i] = -y[i] # posisi dicerminkan keatas batas

vy[i] = -c * vy[i] # kecepatan dibalik dan dikurangi
```

4.2. Syarat Batas *Neumann*

Syarat batas **Neumann** mendefinisikan **nilai turunan (gradien/fluks) fungsi** pada batas domain:

$$\frac{\partial y}{\partial n} = f$$

Di mana n adalah arah normal terhadap batas. Dalam simulasi partikel, ini berkaitan dengan **perubahan kecepatan (percepatan) di batas** — bukan posisi secara langsung, melainkan bagaimana kecepatan berubah saat melewati batas. Sebagai contoh, untuk kasus memantul-teredam, syarat batas Neumann dapat diterapkan pada kecepatan:

$$\frac{\partial v_y}{\partial n} = -c \cdot (vy[i] - vy[i-1])$$

Contoh kode untuk syarat batas Neumann pada gerak partikel:

Kode-2.3

```
# Syarat Batas Neumann - Gradien kecepatan di batas
# Berlaku saat partikel melewati batas bawah (y < 0)
c = 0.8 #koefisien resitusi (0 < c <= 1)
If y[i] < 0:
    y[i] = -y[i] # posisi dicerminkan keatas batas

    vy[i] = -c * (vy[i] - vy[i-1]) # kecepatan dibalik dan dikurangi
```

4.2. Perbandingan *Dirichlet* & *Neumann*

Aspek	Dirichlet	Neumann
Yang dikontrol	Nilai fungsi di batas ($y = f$)	Gradien/turunan di batas ($\partial y / \partial n = f$)
Analogi fisis	Posisi partikel dikunci	Fluks/gaya di batas dikontrol
Contoh implementasi	$y[i] = -y[i]$	$vy[i] = -c \cdot (vy[i] - vy[i-1])$

Kondisi fisik	Dinding keras (elastic wall)	Permukaan dengan gesekan / fluks terkontrol
---------------	------------------------------	---

Catatan: Dalam praktikum ini, syarat batas Dirichlet adalah yang paling umum digunakan karena langsung mengontrol posisi dan kecepatan partikel saat mencapai batas domain. Syarat batas Neumann lebih sering digunakan dalam simulasi medan (*field*), namun juga dapat diterapkan pada partikel untuk mengontrol perubahan kecepatan di batas.

3. Tugas Praktikum

Petunjuk Umum

1. Gunakan bahasa pemrograman Python.
2. Setiap bagian harus disertai (1) grafik hasil simulasi dan (2) analisis singkat (narasi, bukan hanya angka).
3. Jawaban tidak dinilai dari benar–salah angka, tetapi dari (1) ketepatan interpretasi, (2) konsistensi fisika–numerik, (3) kualitas visualisasi

Tugas

1. **Rekonstruksi Hambatan Udara:** Masukkan gaya hambat udara (air drag) $F_d = -kv^2$ ke dalam model gerak peluru. Tuliskan kembali bahasa matematikanya untuk komponen sumbu x dan y, lalu amati bagaimana bentuk lintasan berubah dibandingkan dengan tanpa hambatan udara (efek *ballistic trajectory*).
2. **Analisis Jarak Maksimum:** Gunakan simulasi untuk mencari sudut elevasi yang menghasilkan jarak terjauh (*range*) jika hambatan udara diperhitungkan secara signifikan. Apakah sudut optimalnya tetap berada di 45° ? Jelaskan alasan fisisnya.
3. **Gerak Jatuh Bebas & Terminal Velocity:** Buatlah grafik kecepatan terhadap waktu (v vs t) untuk kasus gerak jatuh bebas dengan hambatan udara. Tunjukkan secara numerik dan grafis bahwa benda akan mencapai **kecepatan terminal** (v_{terminal}) di mana percepatannya menjadi nol.
4. **Visualisasi Komparatif:** Tampilkan dalam satu grafik lintasan peluru dengan tiga variasi koefisien hambat udara (k) yang berbeda.