

**LAPORAN TUGAS AKHIR MATA KULIAH FISIKA
PEMODELAN DAN SIMULASI**

Gerak Harmonik Sederhana pada Sistem Pegas-Massa



Kelompok No.: 6

Tema: Momentum

Anggota Tim:

- 1. Yasin Muhammad Yusuf (A11.2025.16442)**
- 2. Niken Stevia Tribuana (A11.2025.16425)**
- 3. Nayla Khoirul Niswa (A11.2025.16474)**
- 4. Ryan Octovian Felix Ginting (A11.2025.16451)**

Dosen Pengampu:

Dr. Wahyu Aji Eko Prabowo S.Si., M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO
SEMARANG
2025/2026**

LATAR BELAKANG TEORI

Momentum dan Hukum Kekekalan Momentum Momentum merupakan konsep fundamental dalam memahami interaksi antar objek dalam sistem fisik yang lebih luas. Momentum (p) didefinisikan secara matematis sebagai hasil kali antara massa (m) dan kecepatan (v), yang dirumuskan sebagai $p = v \cdot m$. Karena kecepatan merupakan besaran vektor, momentum juga memiliki nilai dan arah. Dalam fenomena tumbukan, berlaku Hukum Kekekalan Momentum Linier yang menyatakan bahwa jumlah momentum sistem sebelum tumbukan sama dengan jumlah momentum setelah tumbukan selama tidak ada gaya luar yang bekerja pada sistem

$$\Sigma p_{awal} = \Sigma p_{akhir}$$

$$m_1 v_1 + m_2 v_2 = m_1 v'_1 + m_2 v'_2$$

Impuls dan Analisis Gaya-Waktu

Impuls (I) didefinisikan sebagai perubahan momentum (Δp) yang dialami oleh suatu benda. Hal ini dihasilkan oleh gaya kontak (F) yang bekerja dalam selang waktu tertentu (Δt). Hubungan antara impuls, gaya, dan momentum sangat penting dalam menganalisis kekuatan interaksi selama tumbukan. Dalam analisis grafik melalui simulasi, nilai impuls dapat dihitung sebagai luas daerah di bawah kurva hubungan antara Gaya (F) terhadap Waktu (t).

Persamaan Impuls:

$$I = F \cdot \Delta t = \Delta p$$

Perubahan Momentum:

$$I = m(v' - v)$$

Koefisien Restitusi (e) dan Jenis Tumbukan

Karakteristik suatu tumbukan ditentukan oleh nilai koefisien restitusi (e), yang merupakan rasio perbandingan kecepatan relatif akhir kedua benda terhadap kecepatan relatif awal. Berdasarkan nilai e , tumbukan diklasifikasikan menjadi tiga:

- Tumbukan Lenting Sempurna ($e = 1$): Energi kinetik dan momentum sistem kekal sepenuhnya.
- Tumbukan Lenting Sebagian ($0 < e < 1$): Momentum tetap kekal, namun terjadi pengurangan energi kinetik karena ditransformasikan menjadi bentuk lain seperti panas atau deformasi.
- Tumbukan Tidak Lenting Sama Sekali ($e = 0$): Karakteristik utamanya adalah kedua benda menyatu setelah benturan dan bergerak dengan kecepatan yang sama.

Persamaan Koefisien Restitusi:

$$e = -\frac{v_2' - v_1'}{v_2 - v_1}$$

Dinamika Tumbukan Dua Dimensi (2D)

Tumbukan 2D melibatkan variabel yang lebih kompleks karena mencakup komponen vektor pada sumbu- x dan sumbu- y . Solusi unik diperoleh dengan mendefinisikan "Garis Dampak" (*Line of Impact*) sebagai garis normal pada titik kontak dan "Bidang Kontak" (*Plane of Contact*) yang tegak lurus terhadap garis tersebut. Dalam model tanpa gesekan, impuls dianggap bekerja hanya di sepanjang garis dampak (sumbu- x), sehingga komponen momentum pada sumbu- untuk setiap objek kekal secara terpisah.

Komponen Kecepatan:

$$\mathbf{v} = v_x \hat{i} + v_y \hat{j}$$

Hukum Kekekalan Sumbu- x :

$$m_1 v_{\{1, i, x\}} + m_2 v_{\{2, i, x\}} = m_1 v_{\{1, f, x\}} + m_2 v_{\{2, f, x\}}$$

PROGRAM SIMULASI YANG DIGUNAKAN

Algoritma Simulasi

Simulasi ini dibangun untuk memodelkan interaksi dua benda dalam ruang satu dimensi (1D) dan dua dimensi (2D). Pengembangan simulasi mengikuti metodologi yang terstruktur dalam tiga tahap utama: pembuatan model, implementasi skenario, dan analisis data. Logika utama program mengikuti alur kerja (*workflow*) sebagai berikut:

1. **Inisialisasi Kondisi Awal:** Menentukan parameter fisik objek seperti massa (m), posisi awal (x, y), diameter, dan kecepatan awal (v). Berdasarkan standar simulasi, objek diatur dengan massa kg dan densitas kg/m^2 pada lingkungan tanpa gesekan (*frictionless*).
2. **Deteksi Tumbukan:** Program secara kontinu memeriksa jarak antar pusat massa kedua objek. Tumbukan terdeteksi secara otomatis jika jarak antar pusat massa lebih kecil atau sama dengan jumlah jari-jari kedua benda.
3. **Perhitungan Fisika (x, y):** Untuk tumbukan 2D, sistem menggunakan koordinat di mana sumbu-x disejajarkan dengan garis dampak (*line of impact*). Garis ini melewati pusat massa kedua objek pada saat kontak.
4. **Kekekalan Momentum dan Restitusi:** Program menghitung vektor kecepatan baru (v) menggunakan Hukum Kekekalan Momentum. Pada model 2D tanpa gesekan, impuls hanya bekerja pada sumbu-x, sehingga komponen momentum pada sumbu-y untuk setiap objek kekal secara terpisah.
5. **Pembaruan Posisi (*State Update*):** Memperbarui posisi benda pada setiap *frame* berdasarkan integrasi kecepatan terhadap waktu (dt)

Implementasi Kode

Implementasi simulasi dilakukan menggunakan bahasa pemrograman Python dengan struktur modular. Modul utama physics.py menangani seluruh logika interaksi fisik, sementara inisialisasi parameter dilakukan pada antarmuka utama. Berikut adalah rincian implementasi kode berdasarkan tahapan simulasi:

A. Inisialisasi Objek

```
1 # ball.py - Ball.__init__() (Baris 42-101)
2 class Ball:
3     def __init__(self, canvas: Tk.Canvas, x_pixels: float, y_pixels: float,
4                   radius_pixels: float, color: str, mass: float, velocity_x: float,
5                   velocity_y: float, pixels_to_meters: float):
6
7         self.canvas = canvas
8         self.radius_pixels = radius_pixels
9         self.radius_meters = radius_pixels * pixels_to_meters
10        self.mass = float(mass)
11        self.pixels_to_meters = pixels_to_meters
12        self.color = color
13
14        # Posisi dan kecepatan dalam satuan meter dan m/s
15        self.position = np.array([x_pixels * pixels_to_meters, y_pixels * pixels_to_meters], dtype=float)
16        self.velocity = np.array([velocity_x, velocity_y], dtype=float)
17
18        # Trail (jejak bola)
19        self.trail_points = deque(maxlen=TRAIL_MAX_LENGTH)
20        self.trail_ids = []
21
22        # Gambar bola di canvas
23        self.canvas_id = canvas.create_oval(
24            x_pixels - radius_pixels,
25            y_pixels - radius_pixels,
26            x_pixels + radius_pixels,
27            y_pixels + radius_pixels,
28            fill=color,
29            outline="black",
30            width=BALL_BORDER_WIDTH
31        )
```

Gambar 1 Inisialisasi Kondisi Awal

B. Deteksi Tumbukan (*Collision Detection*)

```
1 def calculate_collision(ball_1: Ball,
2                        ball_2: Ball,
3                        restitution: float,
4                        contact_tracker: Dict[str, Any]) -> Tuple[float, float]:
5
6     """
7     Deteksi dan tangani tumbukan antara dua bola.
8     Menggunakan Hukum Kekekalan Momentum dan Koefisien Restitusi.
9     """
10
11    # 1. Menghitung jarak antara dua pusat massa (Euclidean Distance)
12    # Rumus: dist = ||pos_1 - pos_2|| = sqrt((x1-x2)^2 + (y1-y2)^2)
13    pos_diff = ball_1.position - ball_2.position
14    dist = np.linalg.norm(pos_diff)
15
16    # Jarak minimal agar bersentuhan: r1 + r2
17    min_dist = ball_1.radius_meters + ball_2.radius_meters
18    f_sample = 0.0
19    finished_impulse = 0.0
20
21    if dist <= min_dist:
```

Gambar 2 Deteksi Tumbukan

C. Perhitungan Vektor Normal dan Kecepatan Relatif Saat tumbukan

```
1 if dist <= min_dist:
2     # 2. Menentukan Normal Vektor (Unit Vector)
3     # Vektor satuan yang menunjuk dari benda 2 ke benda 1
4     # Rumus:  $n = (\text{pos}_1 - \text{pos}_2) / \text{dist}$ 
5     n = pos_diff / (dist + 1e-9)
6
7     # 3. Menghitung Kecepatan Relatif
8     # Kecepatan benda 1 dilihat dari benda 2
9     # Rumus:  $v_{\text{rel}} = v_1 - v_2$ 
10    v_rel = ball_1.velocity - ball_2.velocity
11
12    # 4. Proyeksi Kecepatan Relatif ke Normal Vektor (Dot Product)
13    # Kita hanya peduli pada kecepatan sepanjang Garis Dampak (Line of Impact)
14    # Rumus:  $v_{\text{norm}} = v_{\text{rel}} \cdot n$ 
15    v_norm = np.dot(v_rel, n)
```

Gambar 3 Perhitungan Vektor Normal dan Kecepatan Relatif

D. Kalkulasi Impuls dan Momentum Program menghitung besar impuls

```
1 # 6. Kalkulasi Impuls (Hanya jika benda saling mendekat:  $v_{\text{norm}} < 0$ )
2 if v_norm < 0:
3     # Menghitung Impuls Skalar (j)
4     # Rumus turunan dari Kekekalan Momentum & Restitusi:
5     #  $j = [-(1 + e) * v_{\text{norm}}] / [(1/m_1) + (1/m_2)]$ 
6     j = -(1 + restitution) * v_norm
7     j /= (1.0 / ball_1.mass + 1.0 / ball_2.mass)
8
9     # Mengubah Impuls Skalar menjadi Vektor sepanjang normal
10    # Rumus Impuls Vektor (J):  $J = j * n$ 
11    impulse_vec = j * n
12
13    # 7. Update Kecepatan Akhir (v')
14    # Menerapkan perubahan momentum:  $\Delta p = J$ 
15    #  $v_{\text{new}} = v_{\text{old}} \pm (J / m)$ 
16    ball_1.velocity += impulse_vec / ball_1.mass
17    ball_2.velocity -= impulse_vec / ball_2.mass
18
```

Gambar 4 Kalkulasi Impuls dan Momentum

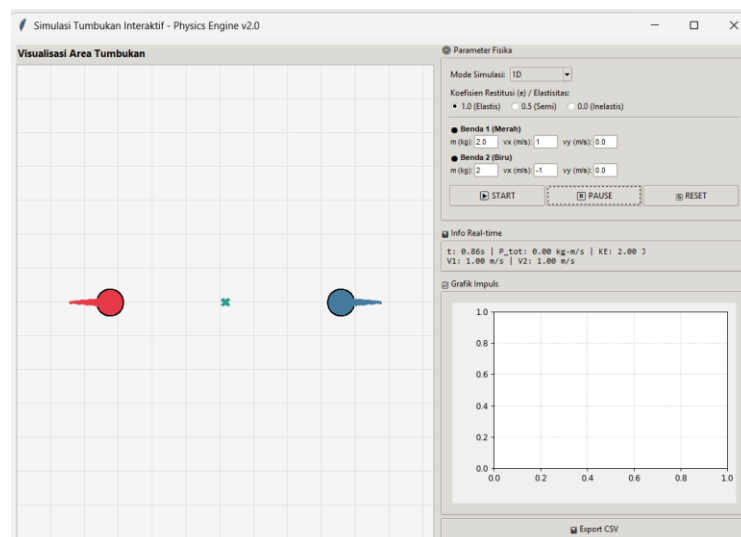
E. .Pembaruan Posisi (*Time Integration*)

```
1 def move(self, time_step: float) -> None:
2     """
3     Gerakkan bola berdasarkan kecepatan dan waktu.
4
5     Parameters:
6     -----
7     time_step : float
8         Delta waktu (detik)
9     """
10    # Update posisi:  $s = s_0 + v \cdot \Delta t$ 
11    self.position += self.velocity * time_step
12
13    # Konversi ke piksel untuk visual
14    pixels_x = self.position[0] / self.pixels_to_meters
15    pixels_y = self.position[1] / self.pixels_to_meters
16
17    # Simpan posisi untuk jejak
18    self.trail_points.append((pixels_x, pixels_y))
19
20    # Update visual
21    self.draw_trail()
22    self.update_visual_position()
```

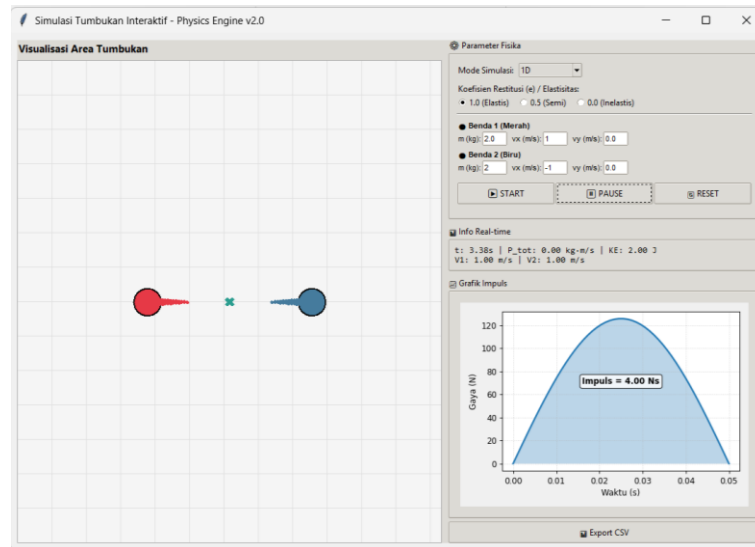
HASIL SIMULASI & ANALISIS

Skenario 1: Tumbukan Lenting Sempurna ($e = 1$)

- **Input:** Massa A = 2 kg, Massa B = 2 kg, Kecepatan Awal saling mendekat.
- **Hasil Simulasi:**



Gambar 5 Lenting sempurna sebelum tumbukan

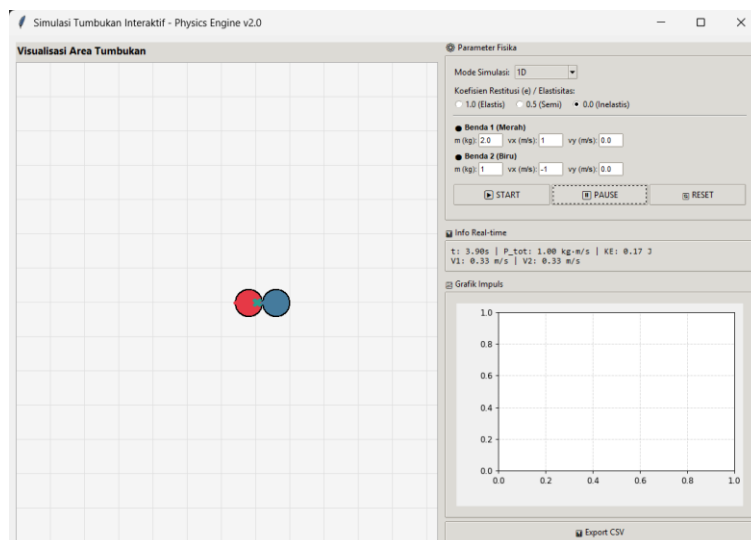


Gambar 6 Lenteng sempurna setelah tumbukan

- **Analisis:** Berdasarkan hasil simulasi, terlihat bahwa setelah tumbukan, kedua benda bertukar kecepatan. Energi kinetik sistem sebelum dan sesudah tumbukan adalah sama (Kekal). Hal ini sesuai dengan teori di mana .

3.2. Skenario 2: Tumbukan Tidak Lenteng Sama Sekali ($e = 0$)

- **Input:** Massa A = 2 kg, Massa B = 1 kg.
- **Hasil Simulasi:**

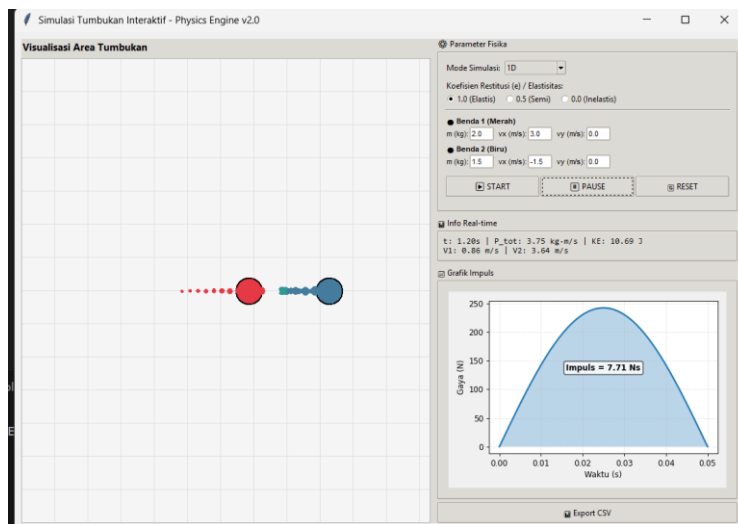


Gambar 7 Tumbukan Tidak Lenteng Sama Sekali

- **Analisis:** Pada simulasi dengan koefisien restitusi , hasil menunjukkan bahwa setelah tumbukan, **kedua benda tidak berpisah**, melainkan bergerak bersama-sama dengan kecepatan yang sama ().

3.3. Analisis Impuls

Dari grafik hubungan Gaya (F) vs Waktu (t) yang dihasilkan simulasi, luas area di bawah kurva menunjukkan nilai Impuls. Nilai ini terbukti sama dengan perubahan momentum (Δp) yang dihitung secara manual:



Gambar 8 Analisis Impuls

DAFTAR PUSTAKA

1. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686-707. <https://arxiv.org/abs/1909.10053>
2. University of Colorado Boulder. (n.d.). *Collision Lab*. PhET Interactive Simulations. Diakses dari <https://phet.colorado.edu/en/simulations/collision-lab>
3. Admin. (2021). *The Development of Physics Teaching Materials based on PhET Simulation to Improve Student Learning Outcomes*. Atlantis Press. <https://www.atlantispress.com/article/126010898.pdf>
4. Fitrianingrum, A. M., & Silangen, P. M. (2025). Utilizing Algodoo for Enhanced Understanding of Collision Concepts in Physics. *Proceedings of the 3rd International Conference on Mathematics and Science Education (ICMSE 2024)*..
5. <https://github.com/Mr-Javkuds/fisika>

Pembagian Bab di laporan:

- I. Latar Belakang Teori/Konsep yang Mendasari
- II. Tuliskan dan Jelaskan Code/Program Simulasi yang Digunakan
- III. Lampirkan Hasil Simulasi dan Lakukan Analisis
- IV. Daftar Pustaka

Aturan Penulisan Laporan:

1. Format Umum

- Laporan diketik menggunakan font Times New Roman, ukuran 12 pt.
- Jarak spasi antar baris menggunakan spasi 1,5.
- Ukuran kertas A4, orientasi portrait.
- Margin disarankan: 4 cm kiri, 3 cm kanan, 3 cm atas, 3 cm bawah.
- Teks ditulis dengan rata kiri-kanan (justify).

2. Batasan Panjang Laporan

- Maksimal panjang laporan adalah 10 halaman.
- Batas 10 halaman ini sudah termasuk:
 - Halaman cover
 - Daftar isi (jika ada)
 - Isi laporan
 - Daftar pustaka / referensi

3. Ketentuan Lain

- Setiap gambar/tabel harus diberi judul dan nomor (misal: Gambar 1. Grafik Perpindahan vs Waktu).
- Sumber referensi (buku/jurnal) wajib dicantumkan di daftar pustaka.
- Plagiarisme tidak diperbolehkan; kutip dan parafrase dengan benar.
- Pengumpulan laporan dan presentasi akan diumumkan kemudian