A11.2021.13656

ILHAM SAFITRO

KOMPUTASI NUMERIK

UAS

A11.64601

dalam Fisika 1

Work and Energy

Seorang pelari berlari sejauh 5 kilometer. Jika ia menghasilkan daya rata-rata 150 watt, berapa energi total yang dikonsumsi oleh pelari tersebut dalam joule? Hitung energi kinetiknya jika ia memiliki massa 60 kg dan berlari dengan kecepatan rata-rata 5 m/s.

Jawaban:

Pekerjaan yang Dilakukan:

dapat dihitung dengan rumus:

$W = P \times t$

Di mana

P adalah daya rata-rata dan

t adalah waktu yang diperlukan. Dalam kasus ini, daya rata-rata adalah 150 watt dan waktu dapat dihitung dari kecepatan dan jarak. Jika kecepatan rata-rata

v adalah 5 m/s, waktu

t adalah 5 km

Konversikan jarak ke meter:

5 km = 5000 m

Maka, t =5000 m

5 m/s = 1000 s

t = 5 m/s

5000 m =1000 s.

Sehingga,

W = 150

W = 150 W × 1000 s=150000 J

Jadi, pelari melakukan pekerjaan sebesar 150000 joule.

Energi Kinetik:

KE dapat dihitung dengan rumus:

$KE = 1/2\ mv2$

Di mana

m adalah massa pelari dan

v adalah kecepatan.

Untuk pelari dengan massa

m = 60 kg dan kecepatan v = 5 m/s

KE = 1/2 × 60 kg × (5m/s)2 = 1/2 × 60 × 25 = 750 J

Jadi, energi kinetik pelari adalah 750 joule.

Kode :

```
# Fungsi untuk menghitung pekerjaan yang dilakukan oleh pelari
def hitung_pekerjaan(daya, waktu):
    pekerjaan = daya * waktu
    return pekerjaan

# Fungsi untuk menghitung energi kinetik pelari
def hitung_energi_kinetik(massa, kecepatan):
    energi_kinetik = 0.5 * massa * (kecepatan ** 2)
    return energi_kinetik

# Konstanta
jarak_km = 5  # jarak dalam kilometer
daya = 150  # daya rata-rata dalam watt
massa_pelari = 60  # massa pelari dalam kg
kecepatan_pelari = 5  # kecepatan rata-rata pelari dalam m/s

# Konversi jarak ke meter
jarak_meter = jarak_km * 1000  # 1 km = 1000 m
```

```
# Hitung waktu yang diperlukan (dalam detik)
waktu = jarak_meter / kecepatan_pelari

# Hitung pekerjaan yang dilakukan oleh pelari
pekerjaan = hitung_pekerjaan(daya, waktu)
print(f"Pekerjaan yang dilakukan oleh pelari: {pekerjaan} Joule")

# Hitung energi kinetik pelari
energi_kinetik = hitung_energi_kinetik(massa_pelari, kecepatan_pelari)
print(f"Energi kinetik pelari: {energi_kinetik} Joule")
```

dalam Fisika 2

Kirchhoff's Law

Soal:

Dalam sebuah rangkaian listrik terdapat tiga simpul dan tiga loop dengan konfigurasi seperti berikut:

Sumber tegangan 1 (V1) sebesar 12V terhubung dengan resistor R1 = 6Ω dan R2 = 4Ω dalam satu loop.

Sumber tegangan 2 (V2) sebesar 9V terhubung dengan resistor R3 = 8Ω dan R2 = 4Ω dalam loop kedua.

Loop ketiga hanya terdiri dari R1 dan R3.

Hitunglah arus yang mengalir melalui masing-masing resistor menggunakan Hukum Kirchhoff dengan penyelesaian komputasi numerik.

Jawaban:

Untuk menyelesaikan masalah ini, kita akan menggunakan persamaan matriks berdasarkan hukum Kirchhoff. Pertama, kita harus menuliskan persamaan untuk setiap loop.

Persamaan Loop:

$V1 - I1R1 - I2R2 = 0$

$V2 - I2R2 - I3R3 = 0$

$I1R1 - I3R3 = 0$

Dimana:

$V1 = 12\ V$

$V2 = 9\ V$

$R1 = 6\ \Omega$

$R2 = 4\ \Omega$

$R3 = 8\ \Omega$

Persamaan Matriks:

$$\begin{bmatrix} R1 & R2 & 0 \\ 0 & R2 & R3 \\ R1 & 0 & -R3 \end{bmatrix} \begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} V1 \\ V2 \\ 0 \end{bmatrix}$$

Masukkan nilai:

$$\begin{bmatrix} 6 & 4 & 0 \\ 0 & 4 & 8 \\ 6 & 0 & -8 \end{bmatrix} \begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} 12 \\ 9 \\ 0 \end{bmatrix}$$

Kode:

```
def gauss_jordan(a, b):
    n = len(b)

    # Augmented matrix
    for i in range(n):
        a[i].append(b[i])

    # Forward elimination
    for i in range(n):
        # Search for maximum in this column
        max_el = abs(a[i][i])
        max_row = i
        for k in range(i+1, n):
            if abs(a[k][i]) > max_el:
```

```
            max_el = abs(a[k][i])
            max_row = k

        # Swap maximum row with current row (column by column)
        for k in range(i, n+1):
            a[max_row][k], a[i][k] = a[i][k], a[max_row][k]

        # Make all rows below this one 0 in current column
        for k in range(i+1, n):
            c = -a[k][i] / a[i][i]
            for j in range(i, n+1):
                if i == j:
                    a[k][j] = 0
                else:
                    a[k][j] += c * a[i][j]

    # Solve equation for an upper triangular matrix
    x = [0 for _ in range(n)]
    for i in range(n-1, -1, -1):
        x[i] = a[i][n] / a[i][i]
        for k in range(i-1, -1, -1):
            a[k][n] -= a[k][i] * x[i]
    return x

# Matriks koefisien
A = [
    [6, 4, 0],
    [0, 4, 8],
    [6, 0, -8]
]
```

```
# Matriks hasil
B = [12, 9, 0]

# Memanggil fungsi gauss_jordan untuk menyelesaikan persamaan
result = gauss_jordan(A, B)

# Menampilkan hasil
I1, I2, I3 = result
print(f"Arus melalui R1 (I1) = {I1:.2f} A")
print(f"Arus melalui R2 (I2) = {I2:.2f} A")
print(f"Arus melalui R3 (I3) = {I3:.2f} A")
```