

Nama : Ulfah Hasanah  
NIM : 1227030036

## Algoritma pemrograman yang digunakan untuk menyelesaikan Soal Fisika Menggunakan Pemrograman Python

Pada praktikum yang dilakukan kami menyelesaikan 3 soal fisika dengan menggunakan pemrograman python. Secara definisi, penyelesaian soal fisika menggunakan pemrograman Python adalah cara untuk memodelkan dan menyelesaikan masalah fisika secara komputasional. Dengan Python, kita dapat menghitung, mensimulasikan, dan memvisualisasikan berbagai fenomena fisika.

Pada program yang pertama, kami melakukan perhitungan penyelesaian jarak fokus lensa ( $f$ ) dengan rumus  $\frac{1}{f} = (n - 1)(\frac{1}{R_1} + \frac{1}{R_2})$  atau invers dari jarak fokus lensa ( $f$ ) yaitu  $f = \frac{1}{(n-1)(\frac{1}{R_1} + \frac{1}{R_2})}$  dengan besar dari indeks bias medium ( $n = 1,50$ ) dan jari jari kelengkungan permukaan lensa  $R_1 = 22$  cm dan  $R_2 = 17,5$  cm. Dalam rumus tersebut, jarak fokus ( $f$ ) dipengaruhi oleh indeks bias lensa ( $n$ ), serta jari-jari kelengkungan dari dua permukaan lensa ( $R_1$  dan  $R_2$ ). Secara matematis, rumus ini menyatakan bahwa fokus lensa adalah kebalikan dari hasil perkalian antara selisih indeks bias dan jumlah kebalikan dari jari-jari kelengkungan. Kemudian kami memasukkan kode program untuk mengolah dari nilai data tersebut. Ketika program dijalankan, program menerima tiga nilai sebagai input, yaitu indeks bias lensa ( $n$ ), jari-jari kelengkungan permukaan pertama ( $R_1$ ), dan jari-jari kelengkungan permukaan kedua ( $R_2$ ). Setelah menerima input, program mulai menghitung selisih indeks bias dengan satu, yaitu  $n - 1$ , karena untuk menunjukkan seberapa besar pengaruh indeks bias terhadap pembentukan fokus. Kemudian, program menghitung kebalikan dari nilai jari-jari kelengkungan kedua permukaan lensa. Artinya, program membagi angka satu dengan masing-masing jari-jari ( $R_1$  dan  $R_2$ ), dan hasilnya dijumlahkan sebagai gambaran tentang kombinasi kelengkungan kedua permukaan lensa. Setelah mendapatkan jumlah kebalikan jari-jari, program mengalikan hasil tersebut dengan selisih indeks bias yang sudah dihitung sebelumnya. Karena yang dicari adalah nilai fokus sebenarnya ( $f$ ), program kemudian membalikkan hasil tersebut untuk mendapatkan nilai fokus yang sesungguhnya ( $1/f$ ). Pada tahap akhir, hasil jarak fokus ini ditampilkan dalam satuan cm yaitu sebesar 19,5 cm. Dengan demikian, algoritma ini

bekerja dengan cara mengambil input, melakukan perhitungan kebalikan dari  $(\frac{1}{R_1} + \frac{1}{R_2})$ , mengalikan hasilnya dengan  $(n - 1)$ , dan akhirnya membalikkan hasil untuk menemukan jarak fokus lensa yaitu dengan hasilnya  $f = 19,5 \text{ cm}$

Pada kode program yang kedua ini kami melakukan perhitungan penyelesaian jarak horizontal maksimum, vertikal maksimum, dan waktu mencapai jarak horizontal maksimum pada gerak parabola menggunakan python. Program ini menghitung dan memvisualisasikan lintasan sebuah benda yang bergerak mengikuti gerak parabola, yaitu gerak melengkung yang terjadi ketika sebuah benda dilempar dengan sudut tertentu terhadap permukaan bumi, dan dipengaruhi oleh gravitasi. Program menggunakan beberapa parameter awal untuk menghitung jarak yang ditempuh benda, ketinggian maksimum yang dicapai, serta waktu yang diperlukan untuk menyelesaikan lintasan tersebut. Pertama, kami menggunakan pustaka `import numpy as np` `import matplotlib.pyplot as plt`. Pustaka `numpy` digunakan untuk melakukan berbagai perhitungan matematis yang diperlukan dalam simulasi ini, sedangkan pustaka `matplotlib` untuk menggambarkan lintasan benda dalam bentuk grafik. Kemudian program akan menginisialisasi beberapa variabel utama: sudut elevasi benda (60 derajat, yang dikonversi ke radian), percepatan gravitasi bumi ( $9.8 \text{ m/s}^2$ ), dan kecepatan awal benda ( $7 \text{ m/s}$ ). Dengan menggunakan rumus trigonometri dasar, kecepatan awal tersebut dipecah menjadi dua komponen: kecepatan horizontal (sepanjang sumbu-x) dan kecepatan vertikal (sepanjang sumbu-y). Kecepatan horizontal tidak berubah selama lintasan, karena tidak ada gaya yang bekerja sepanjang arah tersebut, sementara kecepatan vertikal dipengaruhi oleh gravitasi, yang menarik benda ke bawah. Setelah memecah kecepatan, program menggunakan rumus fisika untuk menghitung beberapa hal penting: jarak horizontal maksimum, ketinggian maksimum, dan waktu yang diperlukan untuk mencapai jarak terjauh. Jarak horizontal maksimum adalah jarak terjauh yang dapat dicapai benda sebelum menyentuh tanah lagi, sementara ketinggian maksimum adalah titik tertinggi yang dicapai selama lintasan. Waktu total menggambarkan berapa lama benda berada di udara sebelum kembali ke tanah. Setelah melakukan perhitungan tersebut, program memulai simulasi gerak benda dengan membagi waktu total menjadi interval kecil, dan menghitung posisi benda (baik horizontal maupun vertikal) pada setiap momen waktu. Pada setiap titik waktu, posisi horizontal benda dihitung menggunakan kecepatan horizontal konstan, sementara posisi vertikal dihitung dengan memperhitungkan percepatan gravitasi yang memperlambat gerak ke atas, dan kemudian mempercepat jatuh ke bawah. Hasil dari perhitungan

jarak horizontal maksimum = 2.16 m, jarak vertikal maksimum = 1.87 m, dan waktu mencapai jarak horizontal maksimum = 1.23 s. Langkah terakhir dalam program adalah memvisualisasikan lintasan benda dalam bentuk grafik. Grafik ini menunjukkan bagaimana benda bergerak dari titik awal, naik hingga mencapai ketinggian maksimum, kemudian jatuh kembali ke tanah setelah mencapai jarak horizontal terjauh. Sumbu horizontal pada grafik menunjukkan jarak (x), dan sumbu vertikal menunjukkan ketinggian (y), sehingga memberikan gambaran visual yang jelas tentang lintasan parabola benda. Setelah selesai melakukan perhitungan dan grafik pada variasi pertama, selanjutnya masuk ke variasi kedua yaitu dengan mengganti sudut elevasi benda menjadi (60 derajat, yang dikonversi ke radian), dan kecepatan awal benda (8 m/s). Program akan mulai bekerja dan akan menampilkan hasil dari perhitungan dan grafik. Hasil perhitungannya jarak horizontal maksimum = 3.26 m, jarak vertikal maksimum = 1.63 m, dan waktu mencapai jarak horizontal maksimum = 1.15 s. Dengan demikian, program ini menggunakan prinsip dasar fisika untuk menghitung dan menggambarkan gerak parabola, mulai dari perhitungan jarak, waktu, dan ketinggian, hingga menampilkan lintasan tersebut dalam bentuk grafik yang mudah dipahami.