

Nama Mahasiswa/NIM	1. Aldit Sheva Osyana (301220075)
Judul Tugas	Simulasi Pelemparan Dadu
Tahun	2024

Simulasi Pelemparan Dadu	
1.	Teori Pendukung
<p>1. Teori Probabilitas Dasar</p> <p>Simulasi pelemparan dadu menggunakan teori dasar probabilitas, di mana setiap sisi dadu memiliki peluang yang sama untuk muncul. Dalam dadu dengan kk sisi, peluang munculnya satu sisi tertentu adalah $1/k$. Teori ini berlaku hanya jika dadu dianggap "adil," artinya setiap sisi memiliki kemungkinan yang sama untuk muncul.</p> <p>Dalam konteks simulasi ini:</p> <ul style="list-style-type: none"> Jika dadu memiliki $k=7$ sisi, peluang munculnya angka tertentu pada setiap pelemparan adalah $1/7$. Menggunakan kode, kita dapat mensimulasikan pelemparan ini secara acak untuk menghitung dan mengestimasi frekuensi dan probabilitas kemunculan angka target. <p>2. Distribusi Uniform (Seragam)</p> <p>Distribusi uniform adalah distribusi di mana setiap hasil memiliki peluang yang sama. Dalam kasus dadu adil:</p> <ul style="list-style-type: none"> Distribusi hasil pelemparan dadu mengikuti distribusi uniform, di mana setiap angka dari 1 hingga kk memiliki peluang yang sama untuk muncul. Hasil simulasi akan menunjukkan distribusi uniform, dengan masing-masing angka muncul dalam jumlah yang hampir sama jika jumlah pelemparan cukup besar. <p>3. Hukum Bilangan Besar</p> <p>Hukum Bilangan Besar menyatakan bahwa saat jumlah percobaan acak semakin banyak, nilai rata-rata hasil percobaan akan mendekati nilai harapan teoretisnya. Dalam simulasi ini:</p> <ul style="list-style-type: none"> Dengan melakukan banyak pelemparan dadu, frekuensi empiris dari masing-masing angka akan mendekati nilai probabilitas teoretisnya, yaitu $1/k$ untuk masing-masing angka pada dadu kk sisi. Hukum ini menjadi penting dalam simulasi karena memungkinkan hasil yang mendekati kondisi ideal dari distribusi probabilitas teoretis. <p>4. Simulasi Monte Carlo</p> <p>Simulasi Monte Carlo adalah metode untuk memperkirakan hasil dari proses acak dengan menggunakan komputasi berbasis acak. Dalam studi kasus pelemparan dadu ini:</p> <ul style="list-style-type: none"> Simulasi Monte Carlo digunakan untuk menghasilkan sampel besar dari pelemparan dadu, memungkinkan kita menghitung hasil statistik (frekuensi dan probabilitas kemunculan angka target). Dengan menyimulasikan banyak percobaan, kita dapat memperkirakan probabilitas dari setiap sisi muncul secara empiris. <p>5. Histogram dan Distribusi Frekuensi</p>	

Histogram adalah alat visualisasi yang efektif untuk melihat distribusi hasil dalam data numerik. Dalam konteks simulasi dadu:

- Histogram frekuensi menunjukkan jumlah kemunculan setiap angka dalam sekumpulan pelemparan dadu.
- Visualisasi ini memungkinkan kita untuk mengamati apakah hasil simulasi mengikuti distribusi yang seragam atau ada ketidaksesuaian.
- Pada simulasi besar (seperti 10.000 pelemparan), histogram akan menunjukkan distribusi yang hampir seragam jika dadu tidak bias.

6. Implementasi di Google Colab dengan Python, NumPy, dan Matplotlib

Google Colab adalah lingkungan interaktif berbasis cloud yang memungkinkan eksekusi kode Python dengan akses ke pustaka yang kuat

seperti NumPy dan Matplotlib. Dalam simulasi ini:

- **NumPy**: Mempermudah operasi numerik pada data array besar dan menyediakan fungsi `np.random.randint` untuk menghasilkan angka acak dalam rentang tertentu, yang merepresentasikan hasil pelemparan dadu.
- **Matplotlib**: Digunakan untuk membuat histogram, memberikan visualisasi yang jelas dari distribusi frekuensi hasil simulasi.
- **Google Colab**: Memfasilitasi pemrosesan data besar dengan cepat, mendukung penggunaan GPU, serta memungkinkan kolaborasi dan penyimpanan otomatis di Google Drive.

7. Prinsip Pemrograman Berbasis Simulasi

Google Colab mendukung metode eksperimen berbasis simulasi, yang dalam kasus ini diterapkan pada simulasi probabilitas. Dengan menjalankan eksperimen yang terprogram, kita bisa:

- Menguji hipotesis probabilitas (misalnya, peluang munculnya sisi tertentu).
- Mengeksplorasi data dan memverifikasi hasil statistik menggunakan eksperimen berulang kali.
- Memahami bahwa hasil simulasi dapat berbeda-beda karena sifatnya yang acak, namun mendekati nilai ekspektasi teoretis dengan jumlah percobaan yang besar.

2.	Alat Dan Bahan
----	----------------

Google colab

3.	Tutorial
----	----------

Langkah 1: Menyiapkan Google Colab

1. Buka Google Colab di colab.research.google.com.
2. Pilih **New Notebook** untuk membuat notebook baru.
3. Rename notebook ini sesuai keinginan, misalnya: "Simulasi Pelemparan Dadu".

Langkah 2: Import Library

Dalam langkah ini, kita akan mengimpor pustaka NumPy untuk operasi numerik dan Matplotlib untuk visualisasi data.

Langkah 3: Menetapkan Parameter Simulasi

Kita akan menetapkan beberapa parameter untuk simulasi ini, seperti jumlah pelemparan dadu, jumlah sisi pada dadu, dan angka target yang ingin kita amati.

Langkah 4: Simulasi Pelemparan Dadu

Selanjutnya, kita akan menggunakan NumPy untuk mensimulasikan hasil pelemparan dadu sebanyak n kali. Kita akan menghasilkan angka acak dari 1 hingga faces.

Langkah 5: Visualisasi Distribusi Hasil Pelemparan

Selanjutnya, kita akan menggunakan Matplotlib untuk membuat histogram yang menunjukkan distribusi hasil pelemparan dadu.

1.	Link Video Tutorial
2.	Referensi:
Intro to Probability and Simulation Sumber ini memberikan dasar teori probabilitas dan konsep simulasi Monte Carlo, yang relevan untuk memahami simulasi pelemparan dadu.	
NumPy Documentation https://numpy.org/doc/2.0/reference/random/generated/numpy.random.randint.html	
Matplotlib Documentation https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html	
Google Colab Guide https://colab.research.google.com/drive/16pBJQePbqkz3QFV54L4NikOn1kwpuRrj	
Hukum Bilangan Besar (Law of Large Numbers) Konsep ini menjelaskan bagaimana hasil rata-rata dari simulasi pelemparan dadu mendekati nilai harapan teoretis seiring bertambahnya jumlah percobaan. <ul style="list-style-type: none">• <i>Sumber: Ross, S. M. (2014). Introduction to Probability and Statistics for Engineers and Scientists.</i>	