|  |  |
| --- | --- |
| Nama Mahasiswa/NIM | 1. Aldit Sheva osyanan (301220075) |
| Judul Tugas | Simulasi lempar koin (RNG) |
| Tahun | 2024 |

|  |  |
| --- | --- |
| Simulasi lempar koin (RNG) | |
|  | **Teori Pendukung** |
| **a. Deskripsi Studi Kasus**  **Lempar koin adalah contoh sederhana dari peristiwa probabilistik. Dalam simulasi ini, kita akan memodelkan lemparan koin yang menghasilkan salah satu dari dua kemungkinan hasil: Head (H) atau Tail (T). Tujuan dari simulasi ini adalah untuk mengamati pola hasil lemparan koin berdasarkan probabilitas yang seimbang (fair coin).**  **Studi ini menggunakan RNG untuk mensimulasikan hasil lemparan koin secara acak. Beberapa pertanyaan yang dapat dijawab melalui simulasi ini adalah:**   * **Berapa probabilitas setiap hasil setelah nnn kali lemparan?** * **Bagaimana distribusi kumulatif hasil lemparan berubah seiring bertambahnya jumlah lemparan?**   **b. Persamaan atau Rumus yang Digunakan**   1. **Probabilitas Dasar:**   **P(Head) = P(Tail) =**   1. **Frekuensi Relatif: Frekuensi relatif dihitung untuk memeriksa apakah hasil mendekati nilai teoretis**  seiring bertambahnya jumlah lemparan:   **F(H) =** , f(T) =   1. **RNG (Randon Number Generator): Simulasi menggunakan fungsi pseudo-random menentukan hasil koin**     **X = RNG(0,1)**  **Jika X < o,5, maka hasil jika tidak, hasil adalah tail**  **c. Pemodelan Menggunakan Python**  **Berikut adalah kode Python untuk mensimulasikan lemparan koin:**  **Python:**  **import numpy as np**  **import matplotlib.pyplot as plt**  **# Parameter simulasi**  **n\_trials = 1000  # Jumlah lemparan koin**  **results = np.random.choice(['Head', 'Tail'], size=n\_trials, p=[0.5, 0.5])**  **# Hitung frekuensi kumulatif**  **heads\_count = np.cumsum(results == 'Head')**  **tails\_count = np.cumsum(results == 'Tail')**  **trials = np.arange(1, n\_trials + 1)**  **# Frekuensi relatif**  **rel\_freq\_heads = heads\_count / trials**  **rel\_freq\_tails = tails\_count / trials**  **# Hasil akhir**  **print(f"Jumlah Head: {heads\_count[-1]}, Jumlah Tail: {tails\_count[-1]}")**  **d. Penjelasan Koding**   1. **np.random.choice: Menghasilkan array hasil simulasi lempar koin berdasarkan probabilitas yang ditentukan.** 2. **np.cumsum: Menghitung jumlah kumulatif dari hasil Head dan Tail setelah setiap lemparan.** 3. **rel\_freq\_heads dan rel\_freq\_tails: Menghitung frekuensi relatif hasil Head dan Tail.** 4. **Visualisasi: Data yang dihasilkan dari simulasi akan divisualisasikan untuk melihat pola distribusi hasil lemparan.**   **e. Tampilan Grafik Pemodelan dan Simulasi**  **Berikut adalah visualisasi grafik dari simulasi:**  **python:**  **# Plot hasil**  **plt.figure(figsize=(10, 6))**  **# Plot frekuensi relatif**  **plt.plot(trials, rel\_freq\_heads, label='Frekuensi Relatif Head', color='blue')**  **plt.plot(trials, rel\_freq\_tails, label='Frekuensi Relatif Tail', color='red')**  **# Tambahkan garis teoretis**  **plt.axhline(y=0.5, color='green', linestyle='--', label='Probabilitas Teoretis (0.5)')**  **# Label dan legend**  **plt.title('Simulasi Lempar Koin dan Frekuensi Relatif')**  **plt.xlabel('Jumlah Lemparan')**  **plt.ylabel('Frekuensi Relatif')**  **plt.legend()**  **plt.grid(True)**  **plt.show()**  **f. Penjelasan Grafik**   1. **Frekuensi Relatif:**    * **Garis biru menunjukkan frekuensi relatif Head.**    * **Garis merah menunjukkan frekuensi relatif Tail.**    * **Kedua garis akan mendekati nilai 0.50.50.5 (probabilitas teoretis) seiring bertambahnya jumlah lemparan.** 2. **Garis Teoretis: Garis hijau putus-putus menunjukkan probabilitas teoretis 0.50.50.5, yang menjadi tolok ukur apakah simulasi konsisten dengan teori.** 3. **Perilaku Stokastik: Pada awal lemparan, frekuensi relatif cenderung fluktuatif, tetapi semakin stabil mendekati nilai teoretis ketika jumlah lemparan meningkat (hukum bilangan besar).** | |
|  | **Alat Dan Bahan** |
| **googlecolab** | |
|  | **Tutorial** |
| 1. **Langkah-langkah:**  * **Buka Google Colab.** * **Buat notebook baru.** * **Salin dan tempelkan kode di atas ke dalam sel-sel kode di Colab.** * **Jalankan setiap sel untuk melihat output dan grafik.**  1. **Langkah-langkah Upload ke Google Colab:**  * **Klik + Code untuk menambahkan sel kode.** * **Klik tombol play di sebelah kiri sel untuk mengeksekusi.**  1. **Grafik Interaktif: Tambahkan %matplotlib inline di bagian atas untuk memastikan grafik ditampilkan langsung di Colab.** | |
|  | **Link Video Tutorial** |
| <https://youtu.be/btz1UNLiOgg?si=wKSa2Ma-4aMo3eNh> | |
|  | **Referensi:** |
| <https://www.khanacademy.org/math/statistics-probability/probability-library> <https://www.geeksforgeeks.org/>  <https://www.w3schools.com/python/module_random.asp>  <https://realpython.com/>  <https://matplotlib.org/stable/contents.html>  <https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html> | |