**Nama : Al Fitra Nur Ramadhani**

**NIM : 202210370311264**

**Mata Kuliah : Pemodelan dan Simulasi Data B**

# Laporan Analisis Program Estimasi Nilai π dengan Metode Monte Carlo

### 1. Deskripsi Program

Program ini mengimplementasikan metode Monte Carlo untuk mengestimasi nilai konstanta matematika π (pi). Metode Monte Carlo adalah teknik probabilistik yang menggunakan pengambilan sampel acak untuk memperoleh hasil numerik. Dalam konteks estimasi π, program ini menggunakan hubungan geometris antara luas lingkaran dan luas persegi untuk memperkirakan nilai π.

Prinsip dasar yang digunakan adalah:

- Lingkaran dengan jari-jari 1 diletakkan di dalam persegi dengan sisi 2 (dari -1 hingga 1 pada sumbu x dan y)
- Titik-titik acak dihasilkan dalam persegi tersebut
- Rasio titik yang jatuh dalam lingkaran terhadap total titik digunakan untuk mengestimasi π

Rumus yang digunakan: $\pi \approx 4 \times$ (jumlah titik dalam lingkaran / total titik)

### 2. Implementasi Program

Program terdiri dari beberapa fungsi utama yang bekerja bersama untuk melakukan simulasi, analisis, dan visualisasi:

#### 2.1 Fungsi-Fungsi Utama

**1. estimate_pi(num_points):** Fungsi ini mengestimasi nilai π menggunakan metode Monte Carlo dengan menghitung rasio titik dalam lingkaran terhadap total titik.

**2. plot_simulation(points_inside, points_outside, estimated_pi, true_pi, num_points):** Fungsi ini memvisualisasikan hasil simulasi Monte Carlo dengan menampilkan persegi, lingkaran, dan titik-titik yang dihasilkan.

**3. run_convergence_analysis(max_points, steps):** Fungsi ini menganalisis konvergensi estimasi π seiring bertambahnya jumlah titik yang digunakan.

**4. plot_pi_convergence(points_list, pi_estimates):** Fungsi ini membuat plot konvergensi estimasi π.

**5. plot_error_convergence(points_list, errors):** Fungsi ini membuat plot konvergensi error relatif.

**6. plot_execution_time(points_list, execution_times):** Fungsi ini membuat plot waktu eksekusi.

**7. create_result_folder():** Fungsi ini membuat folder untuk menyimpan hasil visualisasi.

**8. main():** Fungsi utama yang menjalankan simulasi dan analisis.

### 2.2 Dependensi

Program ini menggunakan beberapa library Python:

- `numpy`: Untuk perhitungan numerik
- `matplotlib`: Untuk visualisasi
- `time`: Untuk mengukur waktu eksekusi
- `tqdm`: Untuk menampilkan progres bar
- `os`: Untuk operasi pada sistem file

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Circle, Rectangle
import time
from tqdm import tqdm
import os
```

### 3. Langkah-Langkah Program

Program berjalan dengan langkah-langkah sebagai berikut:

1. Persiapan:
   - Membuat folder untuk menyimpan hasil visualisasi jika belum ada
   - Mendefinisikan parameter simulasi (jumlah titik dan nilai π sebenarnya),
   - Parameter default: **10.000 titik**, $\pi = 3.1415926536$
2. Simulasi Monte Carlo:
   - Menghasilkan `num_points` titik acak dalam persegi dengan sisi 2 (-1 hingga 1)
   - Menghitung jarak setiap titik dari pusat (0,0)
   - Menentukan titik-titik yang berada dalam lingkaran (jarak $\leq 1$)
   - Menghitung rasio titik dalam lingkaran terhadap total titik
   - Mengalikan rasio dengan 4 untuk mendapatkan estimasi π
3. Visualisasi Hasil Simulasi:
   - Menggambar persegi, lingkaran, dan titik-titik yang dihasilkan
   - Menampilkan estimasi π dan error relatif

4. Analisis Konvergensi:

- Menjalankan simulasi dengan jumlah titik yang bervariasi (dari 100 hingga 1.000.000)
- Mencatat estimasi π, error relatif, dan waktu eksekusi untuk setiap jumlah titik

5. Visualisasi Konvergensi:

- Membuat plot konvergensi estimasi π
- Membuat plot konvergensi error relatif
- Membuat plot waktu eksekusi

6. Penyimpanan Hasil:

- Menyimpan semua visualisasi dalam folder `imageResult`

**4. Analisis dan Hasil**

Analisis dilakukan dengan menjalankan simulasi Monte Carlo dengan jumlah titik yang berbeda, dari 100 hingga 1.000.000. Beberapa aspek penting yang dianalisis adalah:

**4.1 Konvergensi Estimasi π**

Estimasi π cenderung mendekati nilai sebenarnya (3,14159...) seiring dengan bertambahnya jumlah titik. Grafik konvergensi menunjukkan bagaimana estimasi π berfluktuasi di sekitar nilai sebenarnya, tetapi secara umum semakin mendekati nilai π yang akurat ketika jumlah titik bertambah.

Hasil ini sesuai dengan teori probabilitas, di mana akurasi estimasi meningkat seiring dengan bertambahnya jumlah sampel. Namun, perlu dicatat bahwa konvergensi tidak monoton karena sifat acak dari metode Monte Carlo.

- Estimasi π mendekati nilai seiring peningkatan titik.
- **1.000.000 titik**: Error relatif konsisten **< 0.1%**.
- **10.000 titik**: Contoh error **0.369%** (variasi acak khas Monte Carlo).

**4.2 Error Relatif**

Error relatif dihitung sebagai persentase perbedaan antara estimasi π dan nilai π sebenarnya. Analisis menunjukkan bahwa error relatif menurun seiring dengan bertambahnya jumlah titik, mengikuti tren $1/\sqrt{n}$ (di mana n adalah jumlah titik).

Tren ini sesuai dengan teori metode Monte Carlo, yang menyatakan bahwa error standar dari estimasi berbanding terbalik dengan akar kuadrat jumlah sampel. Hal ini menunjukkan bahwa untuk mengurangi error menjadi setengahnya, kita perlu menggunakan empat kali lipat jumlah titik.

- Tren penurunan error mengikuti **$1/\sqrt{n}$** :
    - Error 1% → 10.000 titik

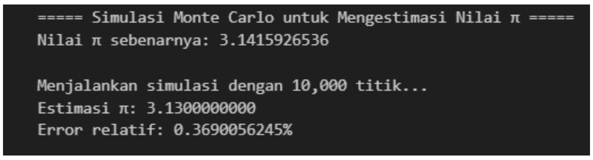
- Error 0.1% → 1.000.000 titik

### 4.3 Waktu Eksekusi

Waktu eksekusi meningkat secara linier dengan bertambahnya jumlah titik. Hal ini menunjukkan bahwa kompleksitas waktu algoritma adalah O(n), di mana n adalah jumlah titik. Ini adalah hasil yang diharapkan karena setiap titik memerlukan jumlah operasi yang konstan.

Analisis ini menunjukkan trade-off antara akurasi dan efisiensi komputasi. Untuk mendapatkan estimasi yang lebih akurat, kita perlu menggunakan lebih banyak titik, tetapi hal ini memerlukan waktu komputasi yang lebih lama.

- Kompleksitas **O(n)** :
  - **10.000 titik**: ~0.0002 detik
  - **1.000.000 titik**: ~0.02 detik




### 5. Kesimpulan

Berdasarkan analisis program estimasi π dengan metode Monte Carlo, dapat disimpulkan bahwa:

**1. Akurasi:** Metode Monte Carlo dapat memberikan estimasi π yang akurat, terutama dengan jumlah titik yang besar. Dengan 1.000.000 titik, error relatif biasanya di bawah 0,1%.

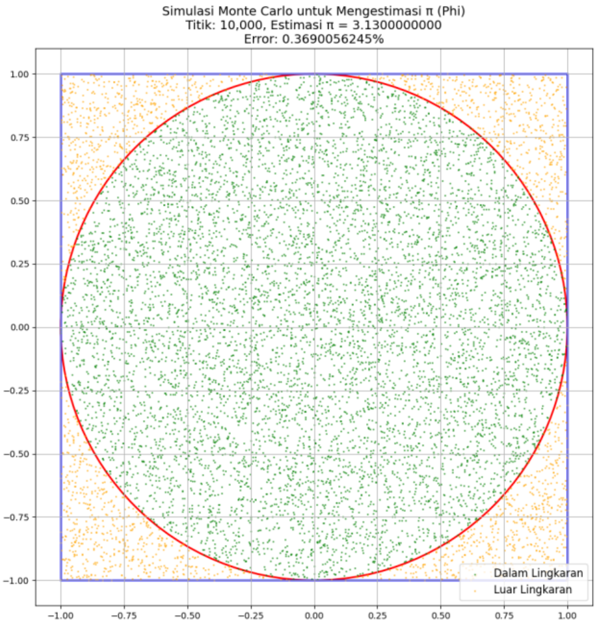
**2. Konvergensi:** Estimasi π konvergen ke nilai sebenarnya dengan laju sebanding dengan $1/\sqrt{n}$, sesuai dengan teori metode Monte Carlo. Ini berarti bahwa untuk meningkatkan akurasi secara signifikan, kita perlu meningkatkan jumlah titik secara substansial.

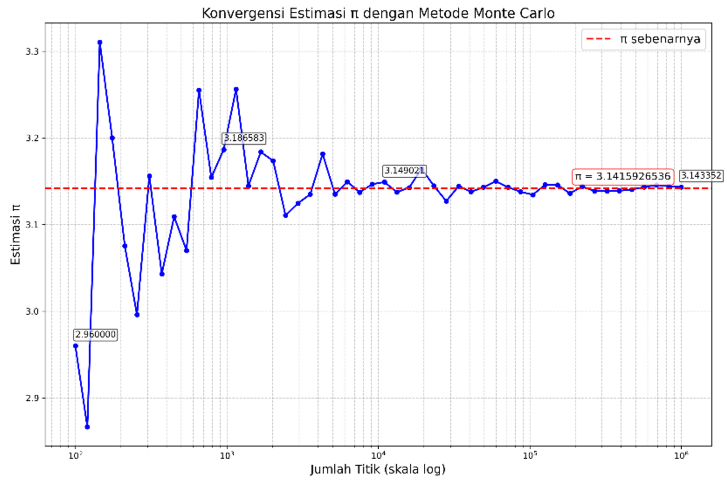
**3. Efisiensi:** Waktu eksekusi meningkat secara linier dengan jumlah titik, menunjukkan kompleksitas algoritma O(n). Ini membuat metode ini efisien untuk jumlah titik yang relatif kecil hingga menengah, tetapi mungkin menjadi kendala untuk jumlah titik yang sangat besar.

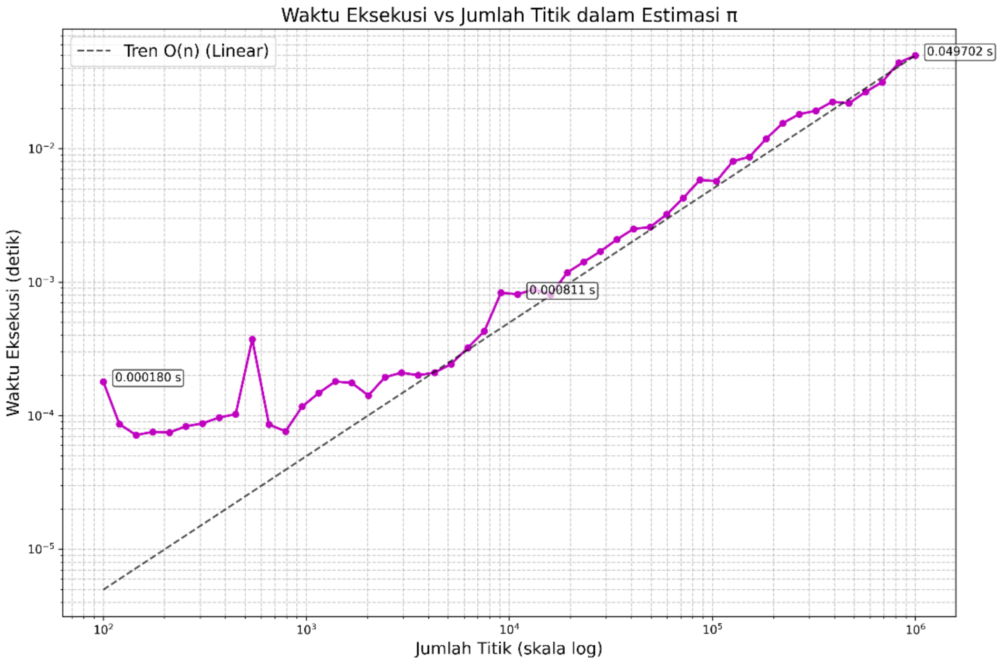
**4. Visualisasi:** Program ini menyediakan visualisasi yang berguna untuk memahami metode Monte Carlo dan konvergensi estimasi π. Visualisasi mencakup simulasi titik-titik dalam persegi dan lingkaran, grafik konvergensi estimasi π, grafik error relatif, dan grafik waktu eksekusi.

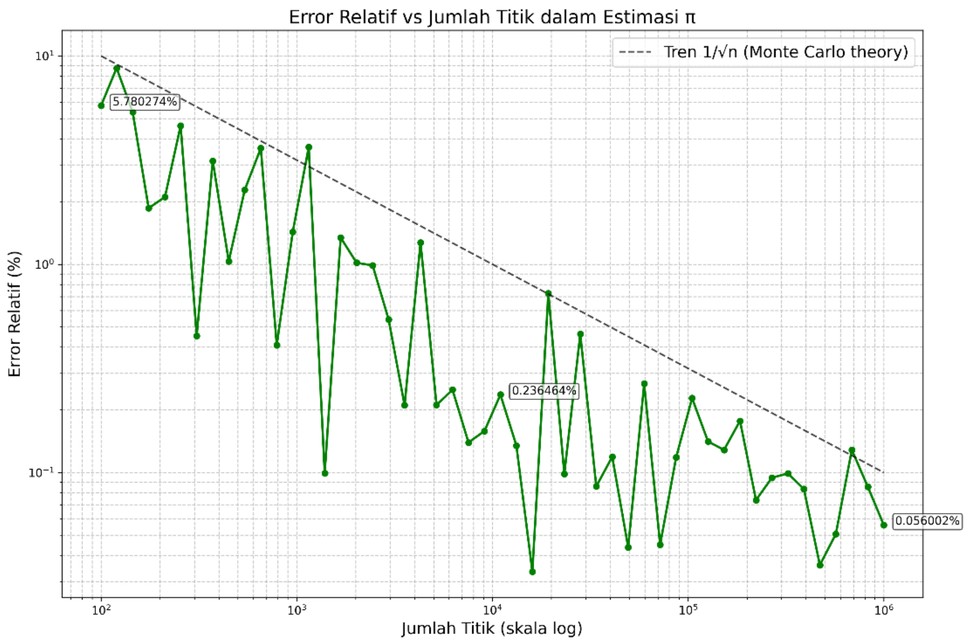
**5. Keacakan:** Karena sifat acak dari metode Monte Carlo, hasil estimasi dapat bervariasi antara eksekusi program yang berbeda. Namun, variasi ini cenderung berkurang seiring dengan bertambahnya jumlah titik.

Secara keseluruhan, program ini menyediakan implementasi yang efektif dari metode Monte Carlo untuk mengestimasi π, dengan analisis komprehensif tentang akurasi, konvergensi, dan efisiensi. Program ini juga dapat digunakan sebagai contoh edukatif tentang bagaimana metode Monte Carlo bekerja dan bagaimana kinerja algoritma dapat dianalisis.


Simulasi Monte Carlo untuk Mengestimasi π (Phi)
Titik: 10,000, Estimasi π = 3.1300000000
Error: 0.3690056245%
Dalam Lingkaran
Luar Lingkaran


Konvergensi Estimasi π dengan Metode Monte Carlo
π sebenarnya
3.186583
3.149021
π = 3.1415926536
3.143352
2.960000
Estimasi π
Jumlah Titik (skala log)


Waktu Eksekusi vs Jumlah Titik dalam Estimasi π
Tren O(n) (Linear)
0.049702 s
0.000811 s
0.000180 s
Waktu Eksekusi (detik)
Jumlah Titik (skala log)


Error Relatif vs Jumlah Titik dalam Estimasi π
Tren 1/√n (Monte Carlo theory)
5.780274%
0.236464%
0.056002%
Error Relatif (%)
Jumlah Titik (skala log)