Analisis Konvergensi Akurasi Estimasi Nilai π menggunakan Metode Simulasi Monte Carlo.

Disusun Oleh:

Irfan Akmal Muzakki (L0224049)

PROGRAM STUDI SAINS DATA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2025

BAB I PENDAHULUAN

1.1 Latar Belakang

Simulasi merupakan cabang ilmu yang penting dalam komputasi dan analisis data, menawarkan kemampuan untuk memodelkan dan menganalisis sistem yang kompleks di dunia nyata atau fenomena matematika yang sulit diselesaikan secara analitis. Dalam banyak kasus, terutama yang melibatkan ketidakpastian (probabilitas), metode simulasi menjadi pendekatan yang paling efisien untuk menghasilkan perkiraan numerik.

Salah satu teknik simulasi yang paling kuat dan serbaguna adalah Metode Monte Carlo. Metode ini memanfaatkan proses sampling acak berulang (iterasi) dalam jumlah besar untuk memperkirakan nilai-nilai yang diinginkan, seperti nilai ekspektasi, probabilitas, atau, dalam kasus ini, konstanta matematika.

Proyek ini bertujuan untuk menerapkan prinsip dasar Simulasi Monte Carlo pada masalah estimasi geometri klasik, yaitu memperkirakan nilai konstanta π (Pi). Dengan memodelkan proses acak (pelemparan titik) dalam suatu batasan geometris, kita dapat menunjukkan efektivitas dan tingkat konvergensi metode Monte Carlo.

1.2 Tujuan Proyek

Tujuan utama dari Laporan Hasil Analisis ini adalah sebagai berikut:

1. Membangun model simulasi berbasis bahasa pemrograman Python untuk mengimplementasikan Metode Monte Carlo.
2. Melakukan simulasi sampling acak untuk memperkirakan nilai konstanta pi dengan menggunakan rasio luas geometris.
3. Menganalisis dan memvisualisasikan tingkat konvergensi estimasi pi terhadap nilai sebenarnya, seiring dengan peningkatan jumlah iterasi yang digunakan.
4. Menghitung dan melaporkan error absolut antara nilai pi hasil simulasi dan nilai pi analitis

BAB II DASAR TEORI

2.1 Konsep Umum Simulasi Monte Carlo

Metode Simulasi Monte Carlo adalah kelas algoritma komputasi yang mengandalkan pengambilan sampel acak berulang (repeated random sampling) untuk mendapatkan hasil numerik. Metode ini sangat berguna untuk memecahkan masalah yang sifatnya deterministik, tetapi terlalu kompleks untuk dihitung secara analitis (misalnya, menghitung integral berdimensi tinggi), atau masalah yang secara intrinsik bersifat stokastik (probabilistik).

Prinsip fundamental metode ini didasarkan pada Hukum Bilangan Besar (Law of Large Numbers). Hukum ini menyatakan bahwa, seiring dengan bertambahnya jumlah percobaan acak, rata-rata hasil dari percobaan tersebut akan konvergen (mendekati) nilai ekspektasi teoritis. Dengan kata lain, semakin banyak titik acak yang dilempar (N), semakin akurat estimasi yang dihasilkan.

2.2 Prinsip Estimasi pi Berbasis Geometri

Dalam konteks proyek ini, Metode Monte Carlo digunakan untuk memperkirakan pi dengan memanfaatkan konsep perbandingan luas.

2.2.1 Setup Geometris

Simulasi dilakukan pada bidang Kartesius dengan *setup* geometris sebagai berikut:

1. Bujur Sangkar Unit (Kotak): Didefinisikan oleh koordinat (x, y) di mana $0 \le x \le 1\ dan\ 0 \le x \le 1$. Luas bujur sangkar ini adalah $A_{kotak} = 1\ x\ 1 = 1$
2. Seperempat Lingkaran (Target): Lingkaran berjari-jari $r = 1$ yang berpusat di titik (0, 0). Karena simulasi terbatas pada kuadran pertama, area targetnya adalah seperempat lingkaran. Luas seperempat lingkaran ini adalah $A_{target} = \frac{1}{4}\boldsymbol{\pi} r^2 = \frac{1}{4}\boldsymbol{\pi}$

2.2.2 Rumus Estimasi

Simulasi melibatkan pelemparan N secara merata ke dalam bujur sangkar unit. Estimasi $\boldsymbol{\pi}$ didapatkan dari rasio antara jumlah titik yang jatuh di dalam seperempat lingkaran (C) dan jumlah total titik yang dilempar (N).

Rasio probabilistik ini mendekati rasio luas:

$$\frac{\text{Titik di Dalam Lingkaran}(C)}{\text{Total Titik}(N)} \approx \frac{\text{Luas Target}}{\text{Luas Kotak}} = \frac{A_{\text{target}}}{A_{\text{kotak}}}$$

$$\frac{C}{N} \approx \frac{\frac{1}{4}\pi}{1}$$

Dengan memanipulasi persamaan tersebut, diperoleh rumus untuk mengestimasi $\boldsymbol{\pi}$:

$$\pi_{\text{estimasi}} = 4 \times \frac{C}{N}$$

2.2.3 Kriteria Penentuan Titik

Untuk setiap titik acak yang dihasilkan (x, y), kriteria yang digunakan untuk menentukan apakah titik tersebut jatuh di dalam seperempat lingkaran (yaitu, masuk dalam hitungan C adalah jika jarak titik tersebut dari pusat (0, 0) kurang dari atau sama dengan jari-jari (1).

Menurut teorema Pythagoras, kuadrat jarak D^2 dari titik (x, y) ke pusat (0,0) adalah:

$D^2 = x^2 + y^2$

Maka, kriteria untuk *Titik di Dalam Lingkaran* adalah:

$$x^2 + y^2 = \leq 1$$

BAB III: METODOLOGI SIMULASI

Komponen	**Spesifikasi**	**Fungsi dalam Proyek**
Bahasa Pemrograman	Python	Bahasa inti yang digunakan untuk mengembangkan algoritma simulasi.
NumPy	*Library* komputasi numerik	Digunakan untuk menghasilkan angka acak seragam dalam jumlah besar secara efisien (np.random.uniform) dan melakukan operasi vektorisasi.
Matplotlib	*Library* visualisasi	Digunakan untuk membuat *Scatter Plot* (visualisasi titik) dan Grafik Konvergensi (analisis akurasi).
Lingkungan Eksekusi	Python *Interpreter* / Terminal	Digunakan untuk menjalankan skrip .py dan menampilkan hasil numerik dan visualisasi *pop-up*.

3.2 Desain Algoritma dan Implementasi

Simulasi Monte Carlo diimplementasikan melalui langkah-langkah prosedural berikut:

Langkah 1: Inisialisasi Parameter
Variabel utama yang dikontrol adalah jumlah total titik yang akan diuji, dilambangkan

sebagai N. Dalam simulasi ini, N ditetapkan sebesar 50.000. Jumlah ini dipilih karena cukup besar untuk mencapai tingkat konvergensi yang memadai terhadap nilai π sebenarnya.

Langkah 2: Generasi Sampel Acak
Menggunakan fungsi dari library NumPy, dihasilkan dua larik (array) dengan ukuran N, yaitu:
1. Larik X, yang berisi N bilangan acak berdistribusi uniform pada interval [0,1] dan merepresentasikan koordinat horizontal.
2. Larik Y, yang berisi N bilangan acak berdistribusi uniform pada interval [0,1] dan merepresentasikan koordinat vertikal.

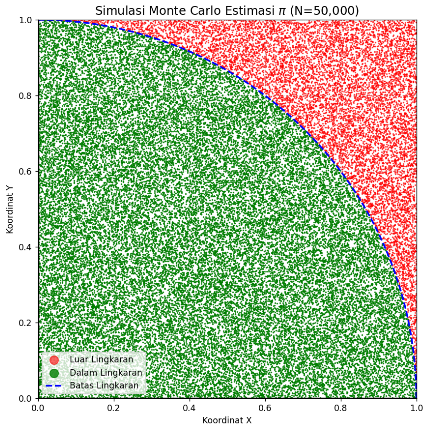
Langkah 3: Penentuan Posisi Titik
Untuk setiap pasangan koordinat (x_i, y_i), dilakukan pengecekan apakah titik tersebut berada di dalam seperempat lingkaran dengan menggunakan kriteria:
$x_i^2 + y_i^2 \leq 1$
Proses ini diimplementasikan menggunakan operasi vektorisasi, sehingga seluruh titik dapat dievaluasi secara simultan. Hasil dari proses ini berupa larik Boolean, dengan nilai True untuk titik di dalam lingkaran dan False untuk titik di luar lingkaran.

Langkah 4: Penghitungan Jumlah Titik dan Estimasi
Jumlah titik yang berada di dalam lingkaran (C) dihitung dengan menjumlahkan seluruh nilai True pada larik Boolean hasil tahap sebelumnya. Selanjutnya, estimasi nilai π dihitung menggunakan persamaan:
$\pi_{estimasi} = 4 \times (C / N)$

BAB IV: ANALISIS DAN HASIL

```
==========================================
          HASIL SIMULASI PI
==========================================
Total Titik (N)          : 50,000
Titik di Dalam Lingkaran (C): 39,303
Estimasi Nilai Pi        : 3.144240
Nilai Pi Sebenarnya      : 3.141593
Error Absolut            : 2.647346e-03
==========================================
PS C:\Users\silvi>
```

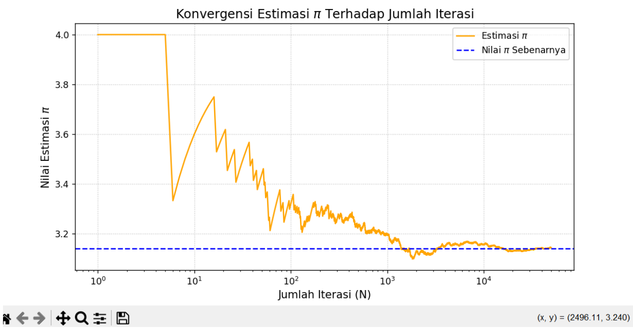
Dari hasil di atas, terlihat bahwa estimasi pi yang diperoleh, yaitu 3.144240, memiliki tingkat akurasi yang tinggi, mendekati nilai sebenarnya dengan error absolut yang sangat kecil, yaitu sekitar 0.002647.



Gambar menyajikan representasi visual dari proses sampling acak yang dilakukan oleh simulasi.

- Titik-titik hijau (39.303 titik) merepresentasikan sampel yang jatuh di dalam seperempat lingkaran dan digunakan dalam perhitungan C.
- Titik-titik merah (10.697 titik) merepresentasikan sampel yang jatuh di luar seperempat lingkaran.

Visualisasi ini secara intuitif menegaskan bahwa rasio kepadatan titik di area target (hijau) terhadap total area (merah + hijau) adalah kunci untuk memperkirakan rasio luas π/4, sesuai dengan prinsip geometri yang dijelaskan dalam Bab II. Distribusi titik yang tampak merata juga mengkonfirmasi kualitas *random number generator* yang digunakan.



Fluktuasi Awal: Pada iterasi awal (skala logaritmik, sumbu X), nilai estimasi pi (garis oranye) menunjukkan fluktuasi yang sangat besar (terlihat nilai bahkan mencapai 4.0). Hal ini wajar karena jumlah sampel yang sedikit belum mampu mewakili populasi secara keseluruhan.

Hukum Bilangan Besar: Seiring peningkatan N yang signifikan (terutama melewati 10^3 iterasi), fluktuasi garis oranye berkurang drastis, dan estimasi mulai stabil. Garis oranye semakin mendekati dan berosilasi di sekitar garis putus-putus biru, yang merepresentasikan nilai pi sebenarnya

Kesimpulan Konvergensi: Perilaku ini secara empiris memvalidasi Hukum Bilangan Besar. Semakin banyak sampel acak yang diikutsertakan (semakin besar N), semakin kecil varians estimasi, yang menghasilkan konvergensi yang stabil dan akurat terhadap nilai teoritis yang dicari. Tingkat konvergensi menunjukkan bahwa untuk mencapai akurasi 3 - 4 digit desimal, diperlukan minimal puluhan ribu iterasi.

BAB V: KESIMPULAN

Berdasarkan implementasi Simulasi Monte Carlo untuk estimasi nilai pi dengan N=50.000 iterasi, proyek ini telah berhasil mencapai tujuan yang ditetapkan dalam Bab I.

5.1 Rangkuman Hasil Utama

1. Validitas Metode: Metode Monte Carlo terbukti efektif dalam memperkirakan konstanta pi melalui pendekatan probabilitas geometri, yang didasarkan pada perhitungan rasio titik acak yang jatuh di dalam seperempat lingkaran terhadap total titik.
2. Tingkat Akurasi: Dengan N=50.000 iterasi, nilai estimasi pi yang diperoleh adalah 3.144240. Hasil ini menunjukkan tingkat akurasi yang tinggi dengan Error Absolut sebesar 2.647346 x 10^{-3}, memvalidasi keandalan metode sampling acak.
3. Konvergensi: Analisis visualisasi konvergensi secara jelas menunjukkan bahwa estimasi pi mengikuti Hukum Bilangan Besar. Nilai estimasi yang sangat fluktuatif pada iterasi awal dengan cepat menstabilkan diri dan berosilasi sangat dekat dengan nilai pi sebenarnya seiring dengan peningkatan jumlah sampel (N).