PERBANDINGAN EFISIENSI ALGORITMA ITERATIF DAN REKURSIF UNTUK MENGHITUNG JUMLAH DIGIT DALAM BILANGAN.

-Tubes AKA-

DISUSUN OLEH

Bramantya Hendrawan Yoga (103012300257)
M. PAKSI PRATAMA (103012300432)

LATAR BELAKANG

- Bagaimana menghitung jumlah digit dalam sebuah bilangan secara efisien menggunakan dua pendekatan algoritma: iteratif dan rekursif.
- Permasalahan ini relevan untuk pemrosesan data numerik, seperti analisis data statistik atau penghitungan ID.

PENJELASAN ALGORITMA

- **Algoritma Iteratif:**
  - Menggunakan perulangan (loop).
  - Variabel berubah hingga memenuhi kondisi penghentian.
- Contoh dalam pseudo-code:

```
count = 0
while n != 0:
    n = n // 10
    count += 1
return count
```

PENJELASAN ALGORITMA

- **Algoritma Rekursif:**
  - Fungsi memanggil dirinya sendiri.
  - Memiliki kasus dasar (base case) untuk menghentikan pemanggilan.
- Contoh dalam pseudo-code:

```
if n == 0:
    return 0
else:
    return 1 + countDigitsRecursive(n // 10)
```

IMPLEMENTASI PROGRAM

- Ditulis dalam bahasa C++.
- Menggunakan pustaka:
  - <chrono> untuk mengukur waktu eksekusi.
  - <cmath> untuk menghasilkan angka besar.
  - <fstream> untuk menyimpan data ke file CSV.
- Pengukuran Waktu:
  - Iteratif: Menggunakan loop untuk menghitung jumlah digit.
  - Rekursif: Menghitung jumlah digit dengan pemanggilan fungsi.

KOMPLEKSITAS WAKTU

Jumlah digit dalam bilangan n dapat dihitung dengan:

$$\lfloor \log_{10}(n) \rfloor + 1.$$

Iteratif:

- Kompleksitas: O(log10(n)).
- Konsisten dan efisien pada input besar.

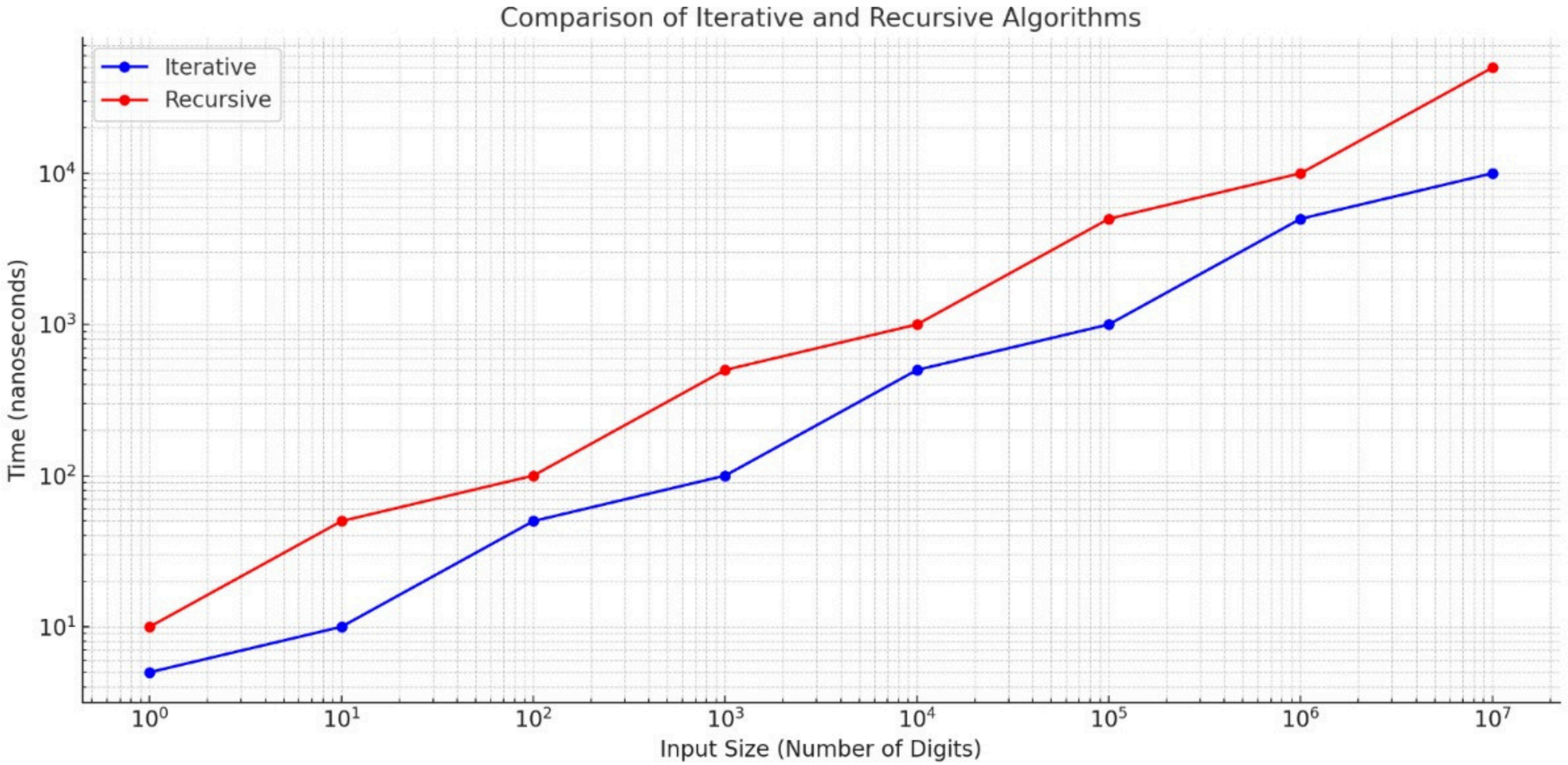
Rekursif:

- Kompleksitas: O(log10(n))
- Lebih lambat karena overhead dari call stack.

HASIL EKSPERIMEN

- Grafik Perbandingan Running Time:
  - **Iteratif**: Kurva landai, menunjukkan efisiensi.
  - **Rekursif**: Kurva lebih curam, menunjukkan peningkatan waktu eksekusi yang tajam.

ANALISIS GRAFIK



OUTPUT PROGRAM

```
Input size: 1, Iterative: 120 ns, Recursive: 70 ns, Digits: 1
Input size: 10, Iterative: 90 ns, Recursive: 110 ns, Digits: 2
Input size: 100, Iterative: 110 ns, Recursive: 70 ns, Digits: 3
Input size: 1000, Iterative: 70 ns, Recursive: 70 ns, Digits: 4
Input size: 10000, Iterative: 50 ns, Recursive: 70 ns, Digits: 5
Input size: 100000, Iterative: 50 ns, Recursive: 70 ns, Digits: 6
Input size: 1000000, Iterative: 50 ns, Recursive: 70 ns, Digits: 7
Input size: 10000000, Iterative: 60 ns, Recursive: 70 ns, Digits: 8
```

PERTIMBANGAN PENGGUNAAN

Iteratif:

- Cocok untuk kasus dengan efisiensi tinggi dan data berskala besar.

Rekursif:

- Cocok untuk kasus divide and conquer yang sederhana.

KESIMPULAN

- Algoritma iteratif lebih efisien dalam hal waktu dan memori.
- Rekursif dapat digunakan jika kebutuhan efisiensi bukan prioritas utama.

REFERENSI

- Munir, R. (2008). Makalah Analisis dan Aplikasi Algoritma.
- Mulyadi, M. (2020). Analisis Algoritma Pencarian.
- Lutfiana, E., et al. (2020). Analisis Kinerja Rekursif dan Iteratif.

THANK YOU