

LAPORAN TUGAS BESAR ANALISIS KOMPLEKSITAS ALGORITMA

**Analisis Perbandingan Algoritma Iteratif dan Rekursif dalam
Menghasilkan Deret Pola Aritmetika**



Disusun oleh:

Anggota:

Adhanafi Ilyasa Sutejo (103012430056)

Mhd Mahathir Gibran Nst (103012400252)

Gusti Arif Fadhillah (103012430059)

**Jurusan Informatika - Fakultas Informatika
Telkom University
2025**

Daftar Isi

1.Pendahuluan.....	3
2.Metode.....	4
2.1 Metode Iteratif.....	4
2.2 Metode Rekursif.....	5
3.Hasil Percobaan dan Grafik Perbandingan.....	6
4.Kesimpulan.....	7

1.Pendahuluan

Makalah ini membahas perbandingan algoritma iteratif dan algoritma rekursif dalam menghasilkan deret pola aritmetika. Tujuan penulisan laporan ini adalah untuk menganalisis perbedaan cara kerja, waktu eksekusi, serta efisiensi penggunaan sumber daya antara kedua algoritma tersebut. Deret pola aritmetika merupakan salah satu konsep dasar dalam matematika dan pemrograman, yang sering digunakan sebagai contoh dalam penerapan algoritma karena memiliki aturan yang sederhana dan terstruktur. Algoritma iteratif adalah algoritma yang menyelesaikan masalah dengan menggunakan perulangan (loop) untuk menghasilkan deret secara bertahap, sedangkan algoritma rekursif menyelesaikan masalah dengan cara memanggil dirinya sendiri hingga mencapai kondisi tertentu. Kedua algoritma ini dapat digunakan untuk menghasilkan deret pola aritmetika dengan hasil akhir yang sama, namun memiliki perbedaan dalam proses eksekusi, penggunaan memori, serta kompleksitas algoritmanya.

Dalam analisis ini, kedua algoritma memiliki tujuan yang sama, yaitu menghasilkan deret pola aritmetika berdasarkan nilai suku awal, beda, dan jumlah suku yang ditentukan. Kondisi awal (precondition) dari kedua algoritma adalah tersedianya nilai suku pertama, beda deret, dan jumlah suku yang akan dihasilkan. Sementara itu, kondisi akhir (postcondition) adalah terbentuknya deret pola aritmetika yang sesuai dengan aturan dan jumlah suku yang diminta. Dengan membandingkan algoritma iteratif dan rekursif dalam konteks yang sama, diharapkan dapat diketahui kelebihan dan kekurangan masing-masing algoritma, baik dari segi performa maupun kemudahan implementasi. Hasil analisis ini diharapkan dapat memberikan pemahaman yang lebih baik dalam memilih algoritma yang tepat untuk menyelesaikan permasalahan tertentu, khususnya dalam penerapan konsep deret pola aritmetika pada pemrograman.

2. Metode

2.1 Metode Iteratif

Metode iteratif merupakan metode penyelesaian masalah yang menggunakan proses perulangan (loop) untuk menghasilkan deret pola aritmetika. Pada metode ini, parameter a digunakan sebagai suku awal deret, b menyatakan beda atau selisih antar suku, dan n merupakan jumlah suku yang akan dihasilkan. Proses perhitungan dimulai dari nilai a , kemudian pada setiap iterasi nilai suku diperbarui dengan menambahkan nilai b untuk memperoleh suku berikutnya. Pada algoritma iteratif, program melakukan perulangan sebanyak n kali yang dikontrol oleh variabel iterasi. Selama jumlah iterasi belum mencapai nilai n , proses perhitungan akan terus berlangsung. Setiap nilai suku yang dihasilkan langsung ditampilkan sebelum melanjutkan ke iterasi berikutnya. Dengan pendekatan ini, deret pola aritmetika dapat dihasilkan secara berurutan dan sistematis tanpa memerlukan pemanggilan prosedur atau fungsi secara berulang.

```
Procedure DeretAritmatikaIteratif(a : integer, b : integer, n : integer)
```

```
Kamus
```

```
    i : integer
```

```
    suku : integer
```

```
algoritma
```

```
    suku <- a
```

```
    for i = 1 to n do
```

```
        output(suku)
```

```
        suku <- suku + b
```

```
    endfor
```

```
endprocedure
```

Time Complexity = $O(n)$

2.2 Metode Rekursif

Metode rekursif merupakan metode penyelesaian masalah yang dilakukan dengan cara memanggil prosedur itu sendiri secara berulang hingga mencapai kondisi berhenti (base case). Pada metode ini, pembentukan deret pola aritmetika dilakukan dengan memproses satu suku pada setiap pemanggilan, kemudian memanggil kembali prosedur tersebut untuk menghasilkan suku berikutnya. Nilai suku selanjutnya diperoleh dengan menambahkan beda (b) pada suku saat ini (a), sementara jumlah suku (n) akan dikurangi pada setiap pemanggilan agar proses rekursi mendekati kondisi berhenti. Jika nilai $n = 0$, prosedur akan menampilkan pesan “Deret Kosong” yang menandakan tidak ada suku yang dihasilkan. Jika nilai $n = 1$, prosedur akan menampilkan nilai a sebagai suku terakhir dan proses rekursi dihentikan. Dengan mekanisme ini,

```
Procedure DeretAritmatikaRekursif(a : integer, b : integer, n : integer)
```

```
  Algoritma
```

```
    if n = 0 then
```

```
      output("Deret Kosong")
```

```
    else if n == 1 then
```

```
      output(a)
```

```
    else
```

```
      output(a)
```

```
      DeretAritmatikaRekursif(a + b, b, n - 1)
```

```
    endif
```

Time Complexity = $O(n)$

3. Hasil Percobaan dan Grafik Perbandingan

a	b	n	Total Waktu Iteratif	Total Waktu Rekursif
1	5	10	4300 ns	4100 ns
1	5	100	25600 ns	42800 ns
1	5	1000	227000 ns	504600 ns
1	5	10000	2234000 ns	4907000 ns



Berdasarkan tabel dan grafik perbandingan waktu eksekusi di atas, dapat diamati bahwa waktu eksekusi algoritma iteratif dan rekursif meningkat seiring dengan bertambahnya jumlah suku (n). Pada nilai n yang kecil, perbedaan waktu eksekusi antara kedua algoritma relatif kecil. Namun, ketika nilai n semakin besar, algoritma rekursif menunjukkan peningkatan waktu eksekusi yang lebih signifikan dibandingkan algoritma iteratif.

4. Kesimpulan

Berdasarkan hasil analisis dan percobaan yang telah dilakukan, dapat disimpulkan bahwa algoritma iteratif dan algoritma rekursif sama-sama mampu menghasilkan deret pola aritmetika dengan benar. Kedua algoritma menunjukkan peningkatan waktu eksekusi seiring bertambahnya jumlah suku (n), yang menandakan bahwa keduanya memiliki kompleksitas waktu **$O(n)$** . Namun demikian, hasil percobaan menunjukkan bahwa algoritma iteratif memiliki waktu eksekusi yang lebih efisien dibandingkan algoritma rekursif, terutama pada nilai n yang besar. Perbedaan ini disebabkan oleh adanya overhead pemanggilan fungsi dan penggunaan stack pada algoritma rekursif, yang menyebabkan waktu eksekusinya menjadi lebih lama. Selain itu, algoritma iteratif juga lebih efisien dalam penggunaan memori karena tidak memerlukan penyimpanan tambahan untuk pemanggilan fungsi. Dengan demikian, dapat disimpulkan bahwa algoritma iteratif lebih direkomendasikan untuk menghasilkan deret pola aritmetika dalam skala besar, sedangkan algoritma rekursif lebih sesuai digunakan untuk permasalahan yang membutuhkan pendekatan rekursi.