

Penjelasan Source Code Untuk Menghitung Nilai Pi Dengan Metode Integrasi Simpson 1/3

1. Ringkasan

Dokumen ini membahas tentang implementasi metode integrasi Simpson 1/3 dalam bahasa pemrograman C++ untuk menghitung nilai integral dari sebuah fungsi matematika. Dokumen ini akan menjelaskan konsep dasar metode Simpson 1/3, implementasi kode dalam bahasa C++, hasil pengujian dengan berbagai nilai n , serta analisis hasil yang meliputi hubungan antara nilai hasil integral, galat, dan waktu eksekusi terhadap nilai n .

2. Konsep

Metode integrasi Simpson 1/3 adalah salah satu metode numerik yang digunakan untuk menghitung nilai integral dari sebuah fungsi pada interval tertentu. Metode ini berdasarkan pada pendekatan geometris, di mana area di bawah kurva fungsi diaproksimasi dengan menggunakan polinomial orde kedua (parabola).

Konsep dasar metode Simpson 1/3 adalah dengan membagi interval integrasi menjadi subinterval-subinterval yang sama lebarnya. Kemudian, di setiap subinterval, nilai fungsi pada titik-titik ujung dan titik tengah subinterval digunakan untuk mengaproksimasi bentuk kurva di dalam subinterval tersebut. Pada metode Simpson 1/3, polinomial orde kedua (parabola) digunakan untuk mengaproksimasi kurva dalam setiap subinterval.

Metode Simpson 1/3 merupakan metode yang akurat dan cepat konvergensi, terutama untuk fungsi yang halus. Namun, metode ini memerlukan jumlah subinterval yang genap untuk memberikan hasil yang benar, dan seringkali lebih lambat daripada metode trapesium untuk fungsi yang tidak halus. Oleh karena itu, pemilihan jumlah subinterval yang tepat sangat penting untuk memastikan hasil integral yang akurat dengan waktu komputasi yang masuk akal.

3. Implementasi Kode

```
#include <iostream>
```

```

#include <iomanip>
#include <chrono>
using namespace std;

// Fungsi yang akan diintegrasikan
double f(double x)
{
    return 4.0 / (1 + x * x);
}

// Fungsi untuk menghitung integrasi menggunakan metode
Simpson 1/3
double simpson13(double a, double b, int n)
{
    if (n % 2 != 0)
    {
        cout << "n harus genap." << endl;
        return -1;
    }

    double h = (b - a) / n;
    double sum = f(a) + f(b);

    for (int i = 1; i < n; i += 2)
    {
        sum += 4 * f(a + i * h);
    }

    for (int i = 2; i < n; i += 2)
    {
        sum += 2 * f(a + i * h);
    }

    return (h / 3) * sum;
}

int main()
{
    double a = 0.0;
    double b = 1.0;

    cout << fixed << setprecision(20); // Set presisi
    tampilan hasil

    for (int n : {10, 100, 1000, 10000}) {
        auto start = chrono::high_resolution_clock::now();
        double result = simpson13(a, b, n);
        auto end = chrono::high_resolution_clock::now();

        chrono::duration<double> duration = end - start;

        if (result != -1)
        {
            cout << "Nilai integral untuk n = " << n << "
            adalah: " << result;
            cout << " (waktu eksekusi: " << duration.count()
            << " detik)" << endl;
        }
    }

    return 0;
}

```

```
}
```

4. Hasil Pengujian

```
PS E:\File Kodingan\tugas metnum integrasi\output> cd 'e:\File Kodingan\tugas metnum integrasi\output'
PS E:\File Kodingan\tugas metnum integrasi\output> & .\integrasi.exe
Nilai integral untuk n = 10 adalah: 3.14159261393921518390 (waktu eksekusi: 0.000000000000000000 detik)
Nilai galat untuk n = 10 adalah: 0.00000003965057793209

Nilai integral untuk n = 100 adalah: 3.14159265358975403615 (waktu eksekusi: 0.000001000000000000 detik)
Nilai galat untuk n = 100 adalah: 0.0000000000000003907985

Nilai integral untuk n = 1000 adalah: 3.14159265358979400418 (waktu eksekusi: 0.000002000000000000 detik)
Nilai galat untuk n = 1000 adalah: -0.000000000000000088818

Nilai integral untuk n = 10000 adalah: 3.14159265358978379012 (waktu eksekusi: 0.000032000000000000 detik)
Nilai galat untuk n = 10000 adalah: 0.000000000000000932587
```

5. Analisis Hasil

Berdasarkan hasil yang telah didapatkan, dapat diketahui bahwa nilai integral pada $n = 10$ adalah 3.14159261393921518390 dengan waktu eksekusi yang sangat kecil hingga mendekati nol. Untuk nilai galatnya, pada $n = 10$ mempunyai nilai galat sebesar 0.00000003965057793209. Lalu, nilai integral pada $n = 100$ adalah 3.14159265358975403615 dengan waktu eksekusi yaitu 0.000001 detik. Untuk nilai galatnya, diperoleh sebesar 0.0000000000000003907985. Kemudian pada $n = 1000$, diperoleh nilai integral sebesar 3.14159265358979400418 dengan waktu eksekusi 0.000002 detik. Untuk nilai galatnya, diperoleh sebesar -0.000000000000000088818. Dan terakhir, pada $n = 10000$, diperoleh nilai dari integral sebesar 3.14159265358978379012 dengan waktu eksekusi 0.000032. Untuk nilai galatnya, diperoleh sebesar 0.000000000000000932587.

Dari hasil tersebut, maka dapat ditarik kesimpulan bahwa semakin besar nilai n -nya, maka semakin besar juga waktu eksekusinya. Namun, semakin besar nilai n -nya tidak membuat nilai galatnya semakin kecil. Karena, nilai galat yang paling kecil ditemukan pada $n = 1000$.