

**PENYELESAIAN PERSAMAAN DIFERENSIAL
MENGUNAKAN METODE RUNGE-KUTTA ORDE KEENAM
DENGAN ALGORITMA PARALEL**

*SOLVING DIFFERENTIAL EQUATION USING THE SIXTH
ORDER RUNGE-KUTTA METHOD WITH PARALLEL
ALGORITHM*

IMAN AL FAJRI



**PROGRAM STUDI MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2021**

**PENYELESAIAN PERSAMAAN DIFERENSIAL
MENGUNAKAN METODE RUNGE-KUTTA ORDE KEENAM
DENGAN ALGORITMA PARALEL**

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister

Program Studi

Magister Matematika

Disusun dan diajukan oleh

IMAN AL FAJRI

kepada

**PROGRAM STUDI MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2021**

TESIS

PENYELESAIAN PERSAMAAN DIFERENSIAL
MENGUNAKAN METODE RUNGE-KUTTA ORDE KEENAM DENGAN
ALGORITMA PARALEL

Disusun dan diajukan oleh

IMAN AL FAJRI
Nomor Pokok H022171002

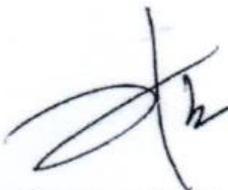
Telah dipertahankan di depan Panitia Ujian Tesis

Pada Tanggal 5 Februari 2021

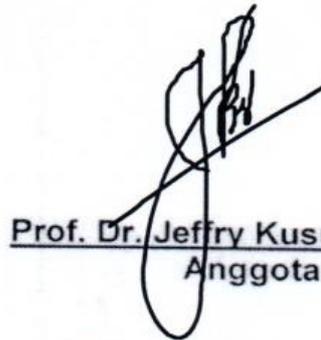
Dan dinyatakan telah memenuhi syarat

Menyetujui

Komisi Penasihat,



Dr. Hendra, S.Si., M.Kom.
Ketua



Prof. Dr. Jeffry Kusuma, Ph.D.
Anggota

Ketua Program Studi
Magister Matematika



Dr. Muhammad Zakir, M.Si.



Dekan Fakultas MIPA
Universitas Hasanuddin

Dr. Eng. Amiruddin, M.Si.

LEMBAR PERNYATAAN KEASLIAN TESIS

Yang bertanda tangan di bawah ini :

Nama : Iman Al Fajri

Nomor induk mahasiswa : H 022 17 1 002

Program studi : Matematika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri dan bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 5 Februari 2021

Yang menyatakan



Iman Al Fajri

PRAKATA

Bismillahirrahmanirrahim. Puji syukur penulis panjatkan ke hadirat Allah Tuhan Yang Maha Kuasa atas karunia dan pertolonganNya sehingga penulis dapat menyelesaikan penelitian dan penyusunan tesis ini sebagai salah satu syarat untuk mencapai gelar Magister pada Program Pascasarjana Universitas Hasanuddin dengan judul “Penyelesaian Persamaan Diferensial Menggunakan Metode Runge-Kutta Orde Keenam dengan Algoritma Paralel”.

Dalam kesempatan ini penulis menyampaikan ucapan terima kasih yang tulus kepada :

1. Kedua orang tua tercinta, ayah dan ibu yang senantiasa memberikan kasih sayang, doa, materi, dan motivasi kepada penulis dalam menjalani pendidikan sampai saat ini.
2. Dr. Hendra, S.Si., M.Kom. dan Prof. Dr. Jeffry Kusuma, Ph.D. selaku pembimbing yang telah meluangkan waktu untuk memberikan bimbingan, semangat, dan arahnya dalam menyelesaikan penulisan tesis ini.
3. Prof. Dr. Moh. Ivan Azis, M.Sc., Prof. Dr. Syamsuddin Toaha, M.Sc., dan Dr. Armin Lawi, M.Eng. selaku tim penguji yang telah banyak memberikan masukan dalam menyempurnakan tulisan ini.

4. Prof. Dr. Syamsuddin Toaha, M.Sc. yang merupakan penasihat akademik yang telah memberikan bimbingan dan arahnya selama menjalani pendidikan di Prodi Magister Matematika.
5. Dr. Muhammad Zakir, M.Si. sebagai Ketua Prodi Magister Matematika yang telah memberikan bantuan dan dukungannya kepada penulis untuk menyelesaikan tulisan ini.
6. Rektor Universitas Hasanuddin dan Direktur Program Pascasarjana beserta seluruh staf yang telah memberikan layanan administrasi yang baik selama penulis menempuh pendidikan di Universitas Hasanuddin.
7. Dekan FMIPA Universitas Hasanuddin, Dr. Eng. Amiruddin, M.Si., seluruh dosen, dan staf administrasi pada program studi Magister Matematika Universitas Hasanuddin yang telah memberikan layanan akademik maupun layanan administrasi selama penulis menempuh pendidikan.
8. Istri tercinta dan kedua mertua yang selalu mendoakan, memberi perhatian, dan semangat walau menunggu agak lama.
9. Keluarga besar yang selalu memberikan doa dan semangat.
10. Rekan-rekan program studi Magister Matematika Universitas Hasanuddin angkatan 2017 dan kepada semua pihak yang telah membantu diucapkan terima kasih yang sebesar-besarnya.

Semoga hasil tesis ini memberikan kontribusi bagi pengembangan ilmu pengetahuan dan memberi informasi ilmiah secara umum.

Makassar, 5 Februari 2021

Iman Al Fajri

ABSTRAK

IMAN AL FAJRI. Penyelesaian Persamaan Diferensial Menggunakan Metode Runge-Kutta Orde Keenam dengan Algoritma Paralel (dibimbing oleh Hendra dan Jeffry Kusuma)

Penelitian tentang paralelisasi terus mengalami perkembangan saat ini, termasuk dalam perhitungan numerik. Pada tulisan ini akan dibahas penyelesaian persamaan diferensial menggunakan metode Runge-Kutta orde keenam dengan algoritma paralel. Makalah ini menyajikan penurunan dari metode Runge-Kutta orde keenam yang cocok untuk implementasi secara paralel. Pengembangan model paralel didasarkan pada struktur ketersebaran Runge-Kutta Tipe kedua yang dibagi ke dalam dua prosesor sehingga diperoleh versi semi implisit yang diselesaikan melalui enam tahapan. Implementasi simulasi paralelisasi menggunakan modul Ray pada bahasa pemrograman Python dengan memberi delay 0.01 detik untuk setiap tahapan yang diselesaikan. Hasil perhitungan dengan model paralel kemudian akan dibandingkan dengan model sekuensial dari sisi akurasi dan waktu eksekusi. Perhitungan numerik menunjukkan bahwa metode paralel lebih mendekati solusi analitik, artinya akurasinya lebih baik. Ditinjau dari sisi waktu eksekusi, metode paralel juga memiliki keunggulan dibandingkan dengan metode sekuensial, yaitu lebih cepat.

Kata kunci : Metode Runge-Kutta Orde Keenam, Algoritma Paralel, Algoritma Sekuensial

ABSTRACT

IMAN AL FAJRI. *Solving Differential Equation Using The Sixth Order Runge-Kutta Method with Parallel Algorithm (mentored by Hendra and Jeffry Kusuma)*

The research on parallelization continues to experience developments at this time, including in numerical calculations. This paper will describe the solution of differential equations using the sixth order Runge-Kutta method with a parallel algorithm. This paper presents a derivation of the sixth order Runge-Kutta method which is suitable for parallel implementation. The development of the parallel model is based on the second type Runge-Kutta sparsity structure which is divided into two processors so that a semi-implicit version is obtained which is completed in six stages. The parallelization simulation implementation uses the Ray module in the Python programming language by giving a delay of 0.01 seconds for each solved stage. The results of calculations with the parallel model will then be compared with the sequential model in terms of accuracy and execution time. The numeric calculation results show that the parallel method means an analytic solution, the accuracy is better. In terms of execution time, the parallel method also has an advantage over the sequential method, which is faster.

Keywords : *sixth order Runge-Kutta method, parallel algorithm, sequential algorithm*

DAFTAR ISI

| | |
|--|-------------------------------------|
| LEMBAR PENGAJUAN | ii |
| LEMBAR PENGESAHAN | Error! Bookmark not defined. |
| LEMBAR PERNYATAAN KEASLIAN..... | iii |
| PRAKATA..... | v |
| ABSTRAK..... | viii |
| ABSTRACT | ix |
| DAFTAR ISI..... | x |
| DAFTAR TABEL..... | xii |
| DAFTAR GAMBAR..... | xiii |
| DAFTAR SINGKATAN..... | xvi |
| BAB I PENDAHULUAN..... | 1 |
| I.1 Latar Belakang | 1 |
| I.2 Rumusan Masalah..... | 4 |
| I.3 Tujuan Penelitian | 4 |
| I.4 Manfaat Penelitian | 4 |
| I.5 Batasan Masalah | 5 |
| BAB II TINJAUAN PUSTAKA..... | 6 |
| II.1 Persamaan Diferensial..... | 6 |
| II.2 Solusi Persamaan Diferensial | 8 |
| II.3 Runge-Kutta Orde Keenam Sekuensial | 8 |
| II.4 Pohon Berakar..... | 11 |
| II.5 Paralelisasi Persamaan Diferensial..... | 13 |
| II.6 Gambaran Umum Komputasi Paralel..... | 14 |
| II.7 Metode Runge-Kutta Paralel..... | 16 |
| II.8 Metode Runge-Kutta Blok Diagonal | 18 |
| II.9 Metode Runge-Kutta Diagonal Implisit..... | 21 |
| II.10 Kondisi Orde..... | 22 |
| II.11 Kinerja Proses Paralel | 24 |
| BAB III METODOLOGI PENELITIAN..... | 26 |

| | | |
|-------------------------|---|-----|
| III.1 | Jenis Penelitian..... | 26 |
| III.2 | Tahapan Penelitian..... | 26 |
| III.3 | Alur Kerja Penelitian | 27 |
| BAB IV | HASIL DAN PEMBAHASAN..... | 29 |
| IV.1 | Metode Runge-Kutta..... | 29 |
| IV.2 | Metode Paralel Tipe II..... | 31 |
| IV.3 | Arsitektur Algoritma untuk Metode Paralel Runge-Kutta Orde Keenam (RK6P2) | 33 |
| IV.4 | Penurunan Metode Runge-Kutta Paralel Orde Keenam | 35 |
| IV.5 | Percobaan Numerik | 59 |
| IV.6 | Beberapa Masalah yang Diuji | 60 |
| IV.7 | Hasil Perhitungan Numerik | 63 |
| IV.8 | Kompleksitas Waktu Algoritma Paralel Metode Runge Kutta Orde Keenam..... | 90 |
| BAB V | PENUTUP | 99 |
| V.1 | Kesimpulan..... | 99 |
| V.2 | Saran..... | 101 |
| DAFTAR PUSTAKA..... | | 102 |
| LAMPIRAN-LAMPIRAN | | 105 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2. 1 Matriks Runge-Kutta dan Graf Berarah | 17 |
| Tabel 2. 2 Jumlah Kondisi Orde..... | 22 |
| Tabel 4. 1 Persamaan Kondisi Orde untuk Metode Runge-Kutta Orde Keenam | 35 |
| Tabel 4. 2 Nilai Galat Global Maksimum Soal 4.1-4.6 antara RK6P2 dan Metode Sekuensial Dibandingkan dengan Solusi Analitik | 63 |
| Tabel 4. 3 Nilai Rata-rata Galat Soal 4.1-4.6 antara RK6P2 dan Metode Sekuensial Dibandingkan dengan Solusi Analitik..... | 64 |
| Tabel 4. 4 Percepatan dan Efisiensi untuk Soal 4.1 | 92 |
| Tabel 4. 5 Percepatan dan Efisiensi untuk Soal 4.2..... | 92 |
| Tabel 4. 6 Percepatan dan Efisiensi untuk Soal 4.3 | 93 |
| Tabel 4. 7 Percepatan dan Efisiensi untuk Soal 4.4 | 93 |
| Tabel 4. 8 Percepatan dan Efisiensi untuk Soal 4.5 | 94 |
| Tabel 4. 9 Percepatan dan Efisiensi untuk Soal 4.6 | 94 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Dua Cara Menggambar Pohon Berakar (Rooted Tree) | 11 |
| Gambar 2. 2 Contoh Pohon Berakar..... | 12 |
| Gambar 2. 3 Graf Dependensi yang Menunjukkan Paralelisme Data | 15 |
| Gambar 2. 4 Graf Dependensi yang Menunjukkan Paralelisme Fungsional | 16 |
| Gambar 2. 5 Bentuk (i) Tabel Butcher Runge-Kutta Orde Keempat dengan Empat Tahap, Dua Prosesor, Dua Paralel | 19 |
| Gambar 2. 6 Bentuk (ii) Tabel Butcher Runge-Kutta Orde Keempat dengan Empat Tahap, Dua Prosesor, Dua Paralel | 19 |
| Gambar 2. 7 Bentuk (iii) Tabel Butcher Runge-Kutta Orde Keempat dengan Empat Tahap, Dua Prosesor, Dua Paralel | 20 |
| Gambar 2. 8 Pohon Berakar (V,E) | 23 |
| Gambar 2. 9 Pohon Berakar dengan Nilai tiap Simpul | 24 |
| Gambar 3. 1 Diagram Alur Kerja Penelitian | 28 |
| Gambar 4. 1 Tabel Butcher..... | 30 |
| Gambar 4. 2 Tabel Butcher 6-tahap Orde Enam..... | 31 |
| Gambar 4. 3 Tabel Butcher 6-tahap Orde Keenam Tipe II | 32 |
| Gambar 4. 4 Struktur Ketersebaran dan Digraf untuk Metode Runge-Kutta Orde Keenam dengan Enam Tahap | 32 |
| Gambar 4. 5 Proses Paralel RK6P2 | 34 |
| Gambar 4. 6 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.1 ($h=0.1$) | 66 |
| Gambar 4. 7 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.1 ($h=0.01$) | 67 |
| Gambar 4. 8 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.1 ($h=0.001$) | 68 |
| Gambar 4. 9 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.1 ($h=0.0001$) | 69 |

| | |
|---|----|
| Gambar 4. 10 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.2 ($h=0.1$) | 70 |
| Gambar 4. 11 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.2 ($h=0.01$) | 71 |
| Gambar 4. 12 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.2 ($h=0.001$) | 72 |
| Gambar 4. 13 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.2 ($h=0.0001$) | 73 |
| Gambar 4. 14 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.3 ($h=0.1$) | 74 |
| Gambar 4. 15 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.3 ($h=0.01$) | 75 |
| Gambar 4. 16 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.3 ($h=0.001$) | 76 |
| Gambar 4. 17 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.3 ($h=0.0001$) | 77 |
| Gambar 4. 18 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.4 ($h=0.1$) | 78 |
| Gambar 4. 19 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.4 ($h=0.01$) | 79 |
| Gambar 4. 20 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.4 ($h=0.001$) | 80 |
| Gambar 4. 21 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.4 ($h=0.0001$) | 81 |
| Gambar 4. 22 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.5 ($h=0.1$) | 82 |
| Gambar 4. 23 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.5 ($h=0.01$) | 83 |
| Gambar 4. 24 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.5 ($h=0.001$) | 84 |

| | |
|---|----|
| Gambar 4. 25 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.5 ($h=0.0001$) | 85 |
| Gambar 4. 26 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.6 ($h=0.1$) | 86 |
| Gambar 4. 27 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.6 ($h=0.01$) | 87 |
| Gambar 4. 28 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.6 ($h=0.001$) | 88 |
| Gambar 4. 29 Hasil Perhitungan dengan Metode Sekuensial, Paralel, dan Metode Analitik Soal 4.6 ($h=0.0001$) | 89 |
| Gambar 4. 30 Diagram Alur Algoritma Paralel Runge-Kutta Orde Keenam | 90 |
| Gambar 4. 31 Waktu Eksekusi Sekuensial dan Paralel Soal 4.1..... | 95 |
| Gambar 4. 32 Waktu Eksekusi Sekuensial dan Paralel Soal 4.2..... | 96 |
| Gambar 4. 33 Waktu Eksekusi Sekuensial dan Paralel Soal 4.3..... | 96 |
| Gambar 4. 34 Waktu Eksekusi Sekuensial dan Paralel Soal 4.4..... | 97 |
| Gambar 4. 35 Waktu Eksekusi Sekuensial dan Paralel Soal 4.5..... | 97 |
| Gambar 4. 36 Waktu Eksekusi Sekuensial dan Paralel Soal 4.6..... | 98 |

DAFTAR SINGKATAN

| | | |
|-------|---|--|
| GGM | : | Galat Global Maksimum |
| SISD | : | Single Instruction Single Data |
| MISD | : | Multiple Instruction Single Data |
| SIMD | : | Single Instruction Multiple Data |
| MIMD | : | Multiple Instruction Multiple Data |
| RK6P2 | : | Runge Kutta Orde Keenam Paralel Dua Prosesor |
| y(Ex) | : | Nilai Perhitungan Analitik |
| yP | : | Nilai Perhitungan Metode Paralel |
| yS | : | Nilai Perhitungan Metode Sekuensial |

BAB I

PENDAHULUAN

I.1 Latar Belakang

Perkembangan penggunaan algoritma paralel saat ini semakin pesat. Hal ini berguna dalam menyelesaikan masalah-masalah yang kompleks dengan dukungan sumber daya yang banyak. Kecepatan eksekusi suatu proses menjadi perhatian yang besar sehingga algoritma sekuensial mulai dianggap kurang efektif.

Algoritma paralel adalah algoritma untuk menyelesaikan masalah numerik. Masalah numerik merupakan salah satu masalah yang memerlukan kecepatan komputasi yang sangat tinggi. Dalam ilmu komputer, sebuah algoritma paralel atau algoritma bersamaan, sebagai lawan sekuensial, merupakan algoritma yang dapat dieksekusi sepotong pada suatu waktu dan pada banyak perangkat pengolahan yang berbeda, kemudian digabungkan bersama-sama lagi sehingga didapatkan hasil yang benar.

Ada tingkatan paralelisme yang bisa digunakan untuk memecahkan persamaan diferensial biasa secara numerik. Caranya adalah dengan restrukturisasi kode manual dan/atau kompilator paralelisasi. Ini dapat ditambah dengan mengganti serial dengan paralel, misalnya, aljabar linier. Namun, level-level paralelisasi ini mungkin tidak diharapkan menghasilkan banyak perbaikan karena banyak kode mungkin masih harus berjalan

secara serial. Pendekatan yang lebih bermanfaat adalah mendesain ulang sifat sekuensial dari algoritma yang digunakan untuk memecahkan persamaan diferensial biasa menjadi algoritma paralel.

Metode Runge-Kutta adalah salah satu metode penyelesaian persamaan diferensial dengan nilai awal secara numerik. Metode ini dapat menghasilkan solusi yang memiliki akurasi yang cukup tinggi dibandingkan dengan metode lainnya.

(Ismail, 2010) membahas tentang metode Runge-Kutta orde keenam yang baru yang tergantung pada metode Runge-Kutta orde kelima yang baru dari (Goeken dan Johnson, 1999). Sifat dari metode ini adalah bahwa dibutuhkan lima evaluasi fungsi di mana metode standar membutuhkan enam atau tujuh evaluasi fungsi. Kemudian metode ini dibandingkan dengan metode Runge Kutta orde kelima yang baru. (Al-Shimmary, 2017) meneliti tentang penyelesaian masalah nilai awal menggunakan Runge Kutta orde keenam. Tujuan utama dari penelitian tersebut adalah untuk menunjukkan representasi baru yang paling sederhana untuk pohon dan turunan metode Runge-Kutta orde keenam dengan tujuh tahap. Perhitungan simbolis digunakan dalam penelitian ini. Dalam rangka menyederhanakan metode, di bagian akhir diberikan contoh untuk menggambarkan metode yang disajikan. (Liu et al, 2011) melakukan penelitian penyelesaian persamaan diferensial dengan Metode Runge-Kutta orde keempat paralel menggunakan dual core. Mereka membandingkan hasilnya dengan metode klasik dan hasilnya

menunjukkan bahwa algoritma numerik paralel memiliki akurasi tinggi dan efisiensi komputasi dalam penggunaan dual core. (Din dan Fudziah Ismail, 2011) melakukan penelitian yang sama namun dengan Metode Runge-Kutta orde kelima. Penelitian mereka terinspirasi dari teori bahwa komputasi paralel mengurangi waktu komputasi. Hasil perhitungan menunjukkan bahwa metode algoritma paralel lebih akurat dan memberikan pengurangan waktu komputasi yang signifikan dibandingkan dengan algoritma sekuensial. (Montejo et al, 2018) membandingkan Metode Runge-Kutta paralel dengan Runge-Kutta sekuensial dan Dorman Prince. Hasilnya adalah waktu komputasi paralel sebanding dengan versi sekuensial. Hanya saja versi paralel butuh sumberdaya komputasi yang jauh lebih banyak. Kemudian mereka mengusulkan algoritma paralel yang baru.

(Butcher, 2009) menulis bahwa sejak munculnya komputer digital, minat baru telah difokuskan pada metode Runge-Kutta, dan sejumlah besar peneliti telah berkontribusi pada perluasan pembahasan menjadi sebuah teori, dan mengembangkan metode tertentu.

Metode Runge Kutta telah berkembang sampai orde tinggi. Algoritma yang digunakan masih sekuensial. Berdasarkan hal tersebut maka penulis terinspirasi untuk meneliti tentang “Penyelesaian Persamaan Diferensial Menggunakan Metode Runge Kutta Orde Keenam dengan Algoritma Paralel”.

I.2 Rumusan Masalah

Berdasarkan pemaparan pada latar belakang, maka permasalahan yang akan dibahas adalah sebagai berikut.

1. Bagaimana konstruksi metode Runge Kutta orde keenam dengan algoritma paralel dalam menyelesaikan persamaan diferensial?
2. Bagaimana perbandingan galat perhitungan metode Runge-Kutta orde keenam paralel dan sekuensial dengan solusi metode eksak?
3. Bagaimana perbandingan waktu eksekusi algoritma sekuensial dan paralel dari metode Runge Kutta orde keenam?

I.3 Tujuan Penelitian

Berdasarkan poin-poin rumusan masalah, maka tujuan penelitian adalah sebagai berikut.

1. Mengonstruksi metode Runge Kutta orde keenam dengan algoritma paralel dalam menyelesaikan persamaan diferensial.
2. Membandingkan galat perhitungan metode Runge-Kutta orde keenam paralel dan sekuensial dengan solusi metode eksak.
3. Membandingkan waktu eksekusi algoritma sekuensial dan paralel dari metode Runge Kutta orde keenam.

I.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan dengan rumusan algoritma paralel metode Runge-Kutta orde keenam dapat meningkatkan

akurasi dan kecepatan perhitungan dalam menyelesaikan persamaan diferensial dengan masalah nilai awal.

I.5 Batasan Masalah

Cakupan permasalahan dalam penelitian ini adalah sebagai berikut.

1. Menyelesaikan persamaan diferensial biasa linear dan nonlinear dengan masalah nilai awal orde satu.
2. Menggunakan metode Runge-Kutta orde keenam dengan algoritma paralel.

BAB II

TINJAUAN PUSTAKA

II.1 Persamaan Diferensial

Persamaan diferensial merupakan kajian yang cukup strategis karena berkaitan dengan bagian-bagian pokok dalam matematika seperti dalam analisis, aljabar, geometri dan bagian pokok lain yang akan sangat berperan dalam pengenalan konsep maupun pemecahan masalah yang berkaitan dengan dunia nyata. Solusi persamaan diferensial dapat ditentukan dengan menggunakan dua metode yaitu metode analitik dan metode numerik. Metode analitik memberikan solusi sejati yaitu solusi yang memiliki galat (error) sama dengan nol sedangkan dengan metode numerik kita memperoleh solusi yang menghampiri solusi sejati. Namun solusi hampiran dapat dibuat seteliti yang diinginkan.

Persamaan diferensial biasa merupakan sebuah bentuk persamaan yang memuat turunan satu atau lebih variabel tak bebas terhadap satu variabel bebas suatu fungsi. Berikut ini adalah contoh persamaan diferensial biasa.

$$\frac{dy}{dx} + y = x^2 \quad (2.1)$$

$$xy \left(\frac{d^3y}{dx^3} \right)^2 + \sin \left(\frac{d^2y}{dx^2} \right) + 8x^2y \left(\frac{dy}{dx} \right) + x^2 = 0 \quad (2.2)$$

Persamaan diferensial biasa terbagi menjadi dua, yaitu persamaan diferensial biasa linear dan persamaan diferensial biasa nonlinear. Bentuk umum persamaan diferensial biasa linear adalah sebagai berikut.

$$F \left[x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^ny}{dx^n} \right] = 0 \quad (2.3)$$

Persamaan (2.3) dapat dituliskan menjadi

$$a_0(x) \frac{d^ny}{dx^n} + a_1(x) \frac{d^{n-1}y}{dx^{n-1}} + \dots + a_n(x)y = b(x) \quad (2.4)$$

di mana $a_0 \neq 0$

Contoh persamaan diferensial biasa linear adalah sebagai berikut.

$$\frac{d^2y}{dx^2} - 3 \frac{dy}{dx} + 2y = 0 \quad (2.5)$$

Persamaan diferensial biasa disebut persamaan diferensial non linear jika persamaan diferensial tersebut tak linear. Contoh persamaan diferensial biasa nonlinear adalah sebagai berikut.

$$\frac{d^2y}{dx^2} - 3 \frac{dy}{dx} + 2y^2 = 0 \quad (2.6)$$

Persamaan diferensial parsial merupakan sebuah bentuk persamaan yang memuat turunan parsial. Satu atau lebih variabel tak bebas terhadap lebih dari satu variabel bebas suatu fungsi.. Salah satu

contoh persamaan diferensial parsial adalah persamaan panas sebagai berikut.

$$\frac{\partial u(x, y, z, t)}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (2.7)$$

Fungsi $u(x, y, z, t)$ digunakan untuk merepresentasikan temperatur pada waktu t pada benda secara fisik dengan koordinat (x, y, z) di mana α adalah difusivitas termal.

II.2 Solusi Persamaan Diferensial

Penyelesaian persamaan diferensial (PD) artinya adalah mencari suatu fungsi $y = f(x)$ yang memenuhi PD tersebut. Fungsi $y = f(x)$ yang memenuhi sebuah PD banyak sekali. Kumpulan fungsi-fungsi yang memenuhi sebuah PD disebut penyelesaian umum persamaan diferensial (PUPD). Salah satu fungsi di dalam kumpulan fungsi-fungsi yang memenuhi sebuah PD disebut penyelesaian khusus PD yang harus memenuhi beberapa kondisi batas.

II.3 Runge-Kutta Orde Keenam Sekuensial

Proses Runge-Kutta (eksplisit) adalah cara memecahkan persamaan diferensial secara numerik (Butcher, 1964). Bentuk umum persamaan diferensial biasa orde satu adalah

$$y' = f(x, y), \quad y(x_0) = y_0 \quad (2.8)$$

yang disebut sebagai *the initial value problem (IVP)* atau persamaan diferensial masalah nilai awal. Diselesaikan di titik $x = x_0 + h$, di mana y, f adalah vektor.

Metode Runge Kutta orde keenam disajikan (Luther, 1968) dalam bentuk umum sebagai berikut.

$$y_{n+1} = y_n + \frac{\{9k_1 + 64k_3 + 49k_5 + 49k_6 + 9k_7\}}{180} \quad (2.9)$$

di mana

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + vh, y_n + vk_1)$$

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{\{4v-1\}k_1 + k_2}{(8v)}\right)$$

$$k_4 = hf\left(x_n + \frac{2h}{3}, y_n + \frac{\{10v-2\}k_1 + 2k_2 + 8vk_3}{27v}\right)$$

$$k_5 = hf\left(x_n + \frac{\left(7 + \frac{(21)^{\frac{1}{2}}}{14}\right)h}{14}, y_n + \{-(77v - 56) + [17v - \right. \quad (2.10)$$

$$8](21)^{\frac{1}{2}}k_1 - 8(7 + (21)^{\frac{1}{2}})k_2 + 48(7 + (21)^{\frac{1}{2}})vk_3 - 3(21 +$$

$$\left. (21)^{\frac{1}{2}}vk_4\}/(392v)\right)$$

$$k_6 = hf \left(x_n + \left(7 - \frac{(21)^{\frac{1}{2}}}{14} \right) h/14, y_n + \{-5([287v - 56] - [59v - 8](21)^{\frac{1}{2}})k_1 - 40(7 - (21)^{\frac{1}{2}})k_2 + 320(21)^{\frac{1}{2}}vk_3 + 3(21 - 121(21)^{\frac{1}{2}})vk_4 + 392(6 - (21)^{\frac{1}{2}})vk_5\}/(1960) \right)$$

$$k_7 = hf \left(x_n + h, y_n + \{15([30v - 8] - [7v(21)^{\frac{1}{2}}])k_1 + 120k_2 - 40(5 + 7(21)^{\frac{1}{2}})vk_3 + 63(2 + 3(21)^{\frac{1}{2}})vk_4 - 14(49 - 9((21)^{\frac{1}{2}})vk_5 + 70(7 + (21)^{\frac{1}{2}})vk_6\}/(180v) \right)$$

Dengan mensubstitusi $v = 1$ ke dalam persamaan, maka akan didapatkan :

$$y_{n+1} = y_n + \frac{\{9k_1 + 64k_3 + 49k_5 + 49k_6 + 9k_7\}}{180}$$

di mana,

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h, y_n + k_1)$$

$$k_3 = hf \left(x_n + \frac{h}{2}, y_n + \{3k_1 + k_2\}/(8) \right)$$

$$k_4 = hf \left(x_n + \frac{2h}{3}, y_n + \frac{\{8k_1 + 2k_2 + 8k_3\}}{27} \right) \quad (2.10a)$$

$$k_5 = hf \left(x_n + \left(7 - \frac{(21)^{\frac{1}{2}}}{14} \right) h/14, y_n + \{3(3(21)^{\frac{1}{2}} - 7)k_1 - 8(7 - (21)^{\frac{1}{2}})k_2 + 48(7 - (21)^{\frac{1}{2}})k_3 - 3(21 + (21)^{\frac{1}{2}})k_4\}/(392) \right)$$

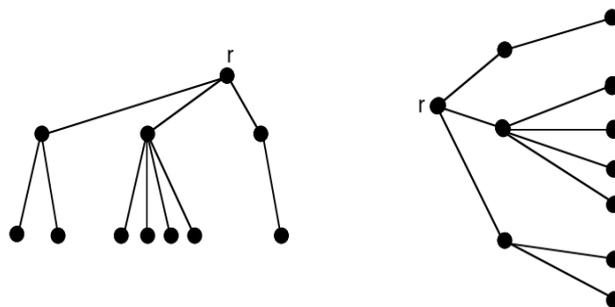
$$k_6 = hf \left(x_n + \left(7 + \frac{(21)^{\frac{1}{2}}}{14} \right) h/14, y_n + \{ -5 (231 + 51(21)^{\frac{1}{2}}) k_1 - \right. \\ \left. 40 (7 + (21)^{\frac{1}{2}}) k_2 - 320(21)^{\frac{1}{2}} k_3 + 3(21 + 121(21)^{\frac{1}{2}}) k_4 + \right. \\ \left. 392(6 + (21)^{\frac{1}{2}}) k_5 \} / (1960) \right)$$

$$k_7 = hf \left(x_n + h, y_n + \{ 15(22 + 7(21)^{\frac{1}{2}}) k_1 + 120k_2 + \right. \\ \left. 40(7(21)^{\frac{1}{2}} - 5) k_3 - 63(3(21)^{\frac{1}{2}} - 2) k_4 - 14(49 + \right. \\ \left. 9(21)^{\frac{1}{2}}) k_5 + 70(7 - (21)^{\frac{1}{2}}) k_6 \} / (180) \right)$$

II.4 Pohon Berakar

Pohon berarah adalah graf berarah yang graf dasarnya adalah pohon. Pohon yang berakar adalah pohon dengan simpul yang ditunjuk yang disebut root. Setiap sisi secara implisit diarahkan menjauh dari root.

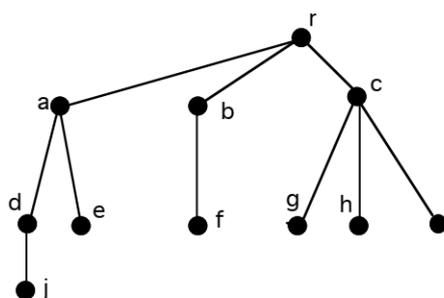
Contoh pohon berakar adalah sebagai berikut.



Gambar 2. 1 Dua Cara Menggambar Pohon Berakar (Rooted Tree)

Dalam pohon berakar, kedalaman atau level dari simpul v adalah jaraknya dari root, yaitu, panjang jalur unik dari root ke v . Dengan demikian, root memiliki kedalaman 0. Ketinggian pohon berakar adalah panjang jalur terpanjang dari akar (kedalaman terbesar di pohon). Jika simpul v langsung mendahului simpul w pada jalur dari root ke w , maka v adalah induk dari w dan w adalah anak dari v . Sedangkan simpul-simpul yang memiliki orangtua yang sama disebut saudara kandung. Simpul w disebut turunan dari simpul v (dan v disebut leluhur w), jika v berada di jalur unik dari root ke w . Jika, sebagai tambahan, $w \neq v$, maka w adalah turunan yang tepat dari v (dan v adalah leluhur yang tepat dari w). Daun pohon berakar adalah setiap simpul yang tidak memiliki anak. Simpul internal pada pohon berakar adalah semua simpul yang memiliki setidaknya satu anak.

Diberikan pohon berakar sebagai berikut.



Gambar 2. 2 Contoh Pohon Berakar

Tinggi pohon ini adalah 3. Kemudian :

- r, a, b, c, dan d adalah simpul internal
- simpul e, f, g, h, i, dan j adalah daun
- simpul g, h, dan i adalah saudara kandung
- simpul a adalah leluhur dari j
- j adalah keturunan dari a

II.5 Paralelisasi Persamaan Diferensial

Tidak setiap masalah perlu diparalelkan untuk menemukan solusinya. Hanya ada dua skenario di mana paralelisasi masuk akal sebagai cara untuk membantu memecahkan masalah yaitu :

1. Masalahnya terlalu besar untuk masuk ke dalam memori satu komputer.
2. dibutuhkan waktu terlalu lama untuk dijalankan.

Sasaran dalam kedua skenario ini dapat digambarkan sebagai pengurangan jumlah waktu (*real time*) yang diperlukan untuk mendapatkan solusinya.

Masalah yang setuju dengan solusi paralel biasanya memiliki beberapa atau semua atribut berikut.

1. Fungsi sisi kanan ODE sulit untuk dievaluasi
2. Interval integralnya panjang
3. Beberapa pengintegralan harus dilakukan
4. Ukuran sistemnya besar

II.6 Gambaran Umum Komputasi Paralel

Komputasi paralel adalah penggunaan komputer paralel untuk mengurangi waktu yang dibutuhkan ketika menyelesaikan satu masalah komputasi. Komputer paralel adalah sistem komputer dengan banyak prosesor yang mendukung pemrograman paralel yang merupakan pemrograman dalam bahasa yang memungkinkan pengembang untuk secara eksplisit menunjukkan bagaimana bagian komputasi yang berbeda dapat dieksekusi secara bersamaan oleh prosesor yang berbeda. (Quinn, 2004).

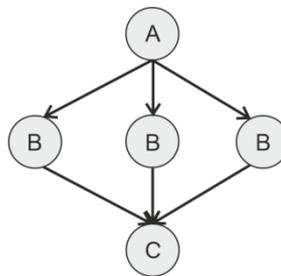
Klasifikasi umum dari komputer paralel dikenal sebagai taksonomi Flynn yang mengklasifikasikan sistem menurut jumlah aliran instruksi dan jumlah aliran data. Arsitektur paling umum diidentifikasi sebagai mesin *single-instruction single-data* (SISD) dan sistem *multiple-instruction multiple-data* (MIMD). Perantara antara sistem SISD dan MIMD adalah *single-instruction multiple-data* (SIMD) dan *multiple-instruction single-data* (MISD). (Pacheco, 1997). Berikut ini adalah perbedaan keempat arsitektur komputer :

1. SISD : model untuk komputasi tunggal.
2. MIMD : model yang melibatkan banyak prosesor secara otonom menjalankan beragam instruksi pada data yang beragam
3. SIMD : model yang melibatkan banyak prosesor secara bersamaan menjalankan instruksi yang sama pada data

yang berbeda

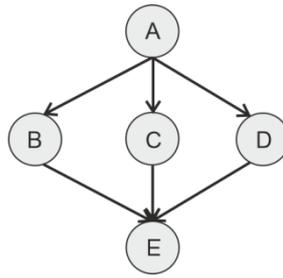
4. MISD : model yang melibatkan banyak prosesor yang menerapkan instruksi berbeda ke satu data. Kemungkinan hipotetis ini umumnya dianggap tidak praktis.

Komputasi paralel dimungkinkan bila ada ketergantungan data dalam komputasi. Ada dua jenis paralelisme yang berlaku jika ada tugas independen yang bekerja pada operasi yang sama ke elemen berbeda dari kumpulan data. Sebuah contoh paralelisme data digambarkan melalui graf ketergantungan data, sejenis graf berarah di mana setiap simpul mewakili tugas yang harus diselesaikan seperti pada gambar 4. (Quinn, 2004).



Gambar 2. 3 Graf Dependensi yang Menunjukkan Paralelisme Data

Paralelisme kedua disebut paralelisme fungsional yang diterapkan ketika ada tugas independen yang mengerjakan operasi berbeda ke elemen data yang berbeda



Gambar 2. 4 Graf Dependensi yang Menunjukkan Paralelisme Fungsional

Paralelisme data disebut juga sebagai paralel domain sedangkan paralelisme fungsional disebut paralel kontrol. Sementara sebagian besar program paralel melibatkan kedua pendekatan, paralelisme data lebih umum dan umumnya lebih mudah. (Pacheco, 1997).

II.7 Metode Runge-Kutta Paralel

Metode Runge-Kutta (Nørsett dan Simonsen, 1987) awalnya dikembangkan untuk perhitungan berurutan. Memang dalam metode eksplisit klasik, perhitungan harus dilanjutkan secara berurutan. Untuk sistem persamaan diferensial biasa

$$y'(x) = f(y), \quad x \geq a \quad (2.11)$$

$$y(a) = y_0, \quad y_0 \in \mathbf{R}^S$$

Metode-Runge Kutta klasik berbentuk

$$k_i = f\left(y_n + h \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad i = 1, 2, \dots, m \quad (2.12)$$

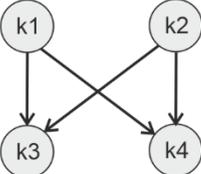
$$y_{n+1} = y_n + h \sum_{i=1}^m b_i k_i \quad (2.13)$$

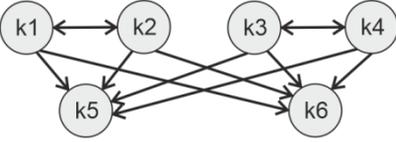
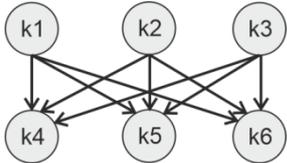
di mana h adalah ukuran langkah. Di sini k_i hanya tergantung pada k_i, \dots, k_{i-1} saja.

Semua nilai k , juga disebut tahapan metode Runge-Kutta, saling bergantung. Dengan menggunakan notasi vektor dan iterasi prediktor-korektor pada metode Runge-Kutta implisit, dapat ditemukan metode dengan orde hingga $2m$, tetapi masih mempertahankan beberapa cara eksplisit.

Iserless dan Nørsett (1990) sudah mendalami metode Runge-Kutta paralel dengan graf berarah (*directed graph*), di mana graf berarah ini mewakili matriks Runge-Kutta untuk menunjukkan kemungkinan metode bisa dievaluasi dengan fungsi paralel. Empat pola yang mereka temukan dapat dilihat di Tabel 2.1.

Tabel 2. 1 Matriks Runge-Kutta dan Graf Berarah

| Tipe | Matriks Runge-Kutta | Graf Berarah |
|------|--|---|
| I | $\begin{pmatrix} x & x & 0 & 0 \\ x & x & 0 & 0 \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{pmatrix}$ |  |
| II | $\begin{pmatrix} x & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ x & x & x & 0 \\ x & x & 0 & x \end{pmatrix}$ |  |

| Tipe | Matriks Runge-Kutta | Graf Berarah |
|------|--|---|
| III | $\begin{pmatrix} x & x & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 \\ x & x & x & x & x & 0 \\ x & x & x & x & 0 & x \end{pmatrix}$ |  |
| IV | $\begin{pmatrix} x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & 0 & x & 0 \\ x & x & x & 0 & 0 & x \end{pmatrix}$ |  |

Metode Runge-Kutta tahap- s adalah R -paralel dan P -prosesor, jika R dan P adalah bilangan bulat terkecil yang pada tahap internal s , metode tersebut dapat dievaluasi dalam satuan waktu R dengan menggunakan prosesor P . (Jackson dan Nørsett, 1995)

II.8 Metode Runge-Kutta Blok Diagonal

Cara biasa untuk mengatasi pembatas orde yang terkait dengan diagonal A dan mempertahankan paralelisasi adalah dengan memungkinkan A menjadi blok diagonal. Blok dapat digunakan untuk membangun sifat yang diinginkan lainnya untuk metode ini (seperti stabilitas A atau orde tinggi) dan masih diproses secara paralel. Tujuan konstruksi yang efisien adalah agar elemen diagonal pada blok yang diberikan sama. Berikut ini adalah metode 4-tahap, 2-paralel, 2-prosesor, dengan orde 4 (Iserles dan Nørsett, 1990).

| | | | | |
|---------------------------|---------------------------|---------------------------|----------------------------|----------------------------|
| $\frac{1}{3}$ | $\frac{1}{3}$ | | | |
| $\frac{2}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | | |
| $\frac{21+\sqrt{57}}{48}$ | | | $\frac{21+\sqrt{57}}{48}$ | |
| $\frac{27-\sqrt{57}}{48}$ | | | $\frac{3-\sqrt{57}}{24}$ | $\frac{21+\sqrt{57}}{48}$ |
| | $\frac{9+3\sqrt{57}}{16}$ | $\frac{9+3\sqrt{57}}{16}$ | $-\frac{1+3\sqrt{57}}{16}$ | $-\frac{1+3\sqrt{57}}{16}$ |

Gambar 2. 5 Bentuk (i) Tabel Butcher Runge-Kutta Orde Keempat dengan Empat Tahap, Dua Prosesor, Dua Paralel

Jika diasumsikan lebih lanjut bahwa blok diagonal mungkin penuh, maka dapat dibangun metode 4-tahap, 2-paralel, 2-prosesor dari orde 4 (Iserles dan Nørsett, 1990) :

| | | | | |
|------------------------|--------------------------|--------------------------|----------------------|-----------------------|
| $\frac{3-\sqrt{3}}{6}$ | $\frac{5}{12}$ | $\frac{1-2\sqrt{3}}{12}$ | | |
| $\frac{3+\sqrt{3}}{6}$ | $\frac{1+2\sqrt{3}}{12}$ | $\frac{5}{12}$ | | |
| $\frac{3-\sqrt{3}}{6}$ | | | $\frac{1}{2}$ | $-\frac{\sqrt{3}}{6}$ |
| $\frac{3+\sqrt{3}}{6}$ | | | $\frac{\sqrt{3}}{6}$ | $\frac{1}{2}$ |
| | $\frac{3}{2}$ | $\frac{3}{2}$ | -1 | -1 |

Gambar 2. 6 Bentuk (ii) Tabel Butcher Runge-Kutta Orde Keempat dengan Empat Tahap, Dua Prosesor, Dua Paralel

Jika diimplementasikan dengan dua prosesor, biaya metode ini sama dengan metode Gauss dua tahap, yang juga menggunakan orde 4 juga,

Secara umum, akan dikategorikan metode Runge Kutta paralel sebagai metode s -tahap, k -paralel, dan l -prosesor, di mana k adalah jumlah blok, l adalah ukuran blok maksimum, dan $s = kl$ adalah jumlah dari tahapan.

Sejauh ini hanya dipertimbangkan metode Runge-Kutta dengan blok diagonal yang sepenuhnya dipisahkan. Jika dimisalkan A menjadi blok segitiga bawah dengan blok diagonal, dapat dibangun metode dari orde 4 dengan metode orde 3 implisit untuk kontrol kesalahan lokal (Iserles dan Nørsett, 1990).

| | | | | |
|---------------|----------------|---------------|---------------|---------------|
| $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| $\frac{2}{3}$ | | $\frac{2}{3}$ | | |
| $\frac{1}{2}$ | $-\frac{5}{2}$ | $\frac{5}{2}$ | $\frac{1}{2}$ | |
| $\frac{1}{3}$ | $-\frac{5}{3}$ | $\frac{4}{3}$ | | $\frac{2}{3}$ |
| | -1 | $\frac{3}{2}$ | -1 | $\frac{3}{2}$ |

Gambar 2. 7 Bentuk (iii) Tabel Butcher Runge-Kutta Orde Keempat dengan Empat Tahap, Dua Prosesor, Dua Paralel

Pasangan tahap pertama dapat dihitung secara bersamaan; maka pasangan tahap kedua dapat dihitung secara bersamaan, dalam hal ini menggunakan faktorisasi LU yang sama dengan pasangan pertama.

Ada pembatas urutan pada metode Runge Kutta blok-diagonal yang dapat dibuktikan.

Teorema : Misalkan $\lambda_1, \lambda_2, \dots, \lambda_n$ menjadi koefisien diagonal yang berbeda dari A dengan masing-masing banyaknya adalah $\mu_1, \mu_2, \dots, \mu_n$. Maka urutan p dari setiap paralel $k - paralel$, prosesor-paralel *Differential Iteration Runge Kutta* (DIRK) memenuhi

$$p \leq 1 + \sum_{i=1}^n \min(\mu_i, k) \quad (2.14)$$

Jadi urutan maksimum dari setiap $k - paralel$, metode *Sequential Differential Iteration Runge Kutta* (SDIRK)-prosesor ($\lambda_1 = \lambda_2 = \dots = \lambda_n$) adalah $k + 1$.

II.9 Metode Runge-Kutta Diagonal Implisit

Metode numerik implisit selalu menjadi pilihan ketika menyelesaikan persamaan diferensial biasa yang kaku. Metode Runge-Kutta implisit diagonal yang merupakan bentuk dari metode Runge Kutta semi implisit telah dianggap sebagai metode yang memiliki kelebihan yang hampir sama dengan metode Runge-Kutta implisit, terutama dalam kriteria stabilitasnya. Kadang-kadang metode ini disebut sebagai Runge-Kutta tunggal diagonal implisit, dengan metode Runge-Kutta diagonal implisit tidak harus memiliki diagonal yang sama. (Butcher, 2008)

Metode Runge-Kutta diagonal implisit telah mendapatkan perhatian serius dari banyak peneliti untuk mempelajari subjek tersebut. Banyak jurnal telah dipublikasikan tentang metode ini seperti Nørsett dan Thomsen (1986), Iserless dan Nørsett (1990), Al Rabeh (1993), Jackson

dan Nørsett (1995), Butcher dan Chen (2000), Shirobokov (2002), Kværno (2004) dan Skvortsov (2006).

II.10 Kondisi Orde

Untuk mendapatkan metode Runge-Kutta urutan p , maka perlu mengacu pada kondisi urutan metode. Ada dua cara untuk mengetahui kondisi urutan metode Runge-Kutta. Teknik pertama adalah dengan menggunakan ekspansi Taylor yang memperkenalkan metode dengan cara "alami" di mana kondisi urutan terdiri dari pencocokan perluasan solusi yang dihasilkan oleh satu langkah metode Runge-Kutta dengan perluasan Taylor dari solusi yang tepat. Perhitungan seperti itu terkenal berat dan membosankan, terutama jika diperlukan urutan tinggi (Lambert, 1991).

Untuk meringankan komplikasi berlebihan yang mungkin terjadi dengan menggunakan ekspansi Taylor, pendekatan yang lebih ringan yang merupakan teknik kedua dalam menurunkan metode Runge-Kutta dipelopori oleh Butcher (1987, 2008) di mana kondisi orde ditetapkan menggunakan konsep pohon berakar. Teramati bahwa jumlah kondisi orde meningkat seiring dengan naiknya orde. (Burrage, 1995)

Tabel 2. 2 Jumlah Kondisi Orde

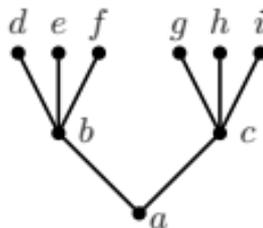
| | | | | | | | | | | |
|---------|---|---|---|---|----|----|----|-----|-----|------|
| Orde | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Kondisi | 1 | 2 | 4 | 8 | 17 | 37 | 85 | 200 | 486 | 1205 |

Suatu pohon berakar (*rooted tree*) t , didefinisikan dengan $r(t)$ yaitu orde dari t , sebagai jumlah simpul pada t . Misal $A(t)$ menunjukkan kelompok automorfisme pada pelabelan t tertentu. Artinya $A(t)$ adalah himpunan pemetaan $\varphi : V \rightarrow V$ sedemikian rupa sehingga $[x, y] \in E$ jika dan hanya jika $[\varphi(x), \varphi(y)] \in E$. Grup $A(t)$ kemudian disebut sebagai grup simetri atau biasa disebut simetri saja yang dilambangkan dengan $\sigma(t)$. Kepadatan dari t yaitu $\gamma(t)$ didefinisikan sebagai perkalian seluruh simpul dari subpohon berakar pada simpul tersebut. Diberikan suatu pohon (V, E) dengan sembilan simpul sebagai berikut.

$$V = \{a, b, c, d, e, f, g, h, i\}$$

$$E = \{(a, b), (a, c), (b, d), (b, e), (b, f), (c, g), (c, h), (c, i)\}$$

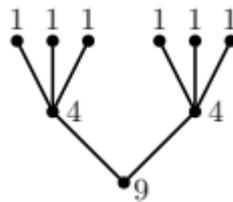
Gambar yang merepresentasikan pohon ini, dengan simpul telah diberi label adalah



Gambar 2. 8 Pohon Berakar (V,E)

Nilai dari $r(t)$ adalah 9. Grup simetri adalah himpunan permutasi yang dihasilkan oleh semua anggota grup simetris di $\{d, e, f\}$, oleh semua anggota grup simetris di $\{g, h, i\}$, dan grup yang dihasilkan oleh permutasi

tunggal , dimana b dan c dipertukarkan, d dan g dipertukarkan, e dan h dipertukarkan, dan f dan i dipertukarkan. Jadi orde grup simetri adalah $\sigma(t) = 3!3!2! = 72$. Sedangkan untuk menghitung $\gamma(t)$, pasang bilangan bulat ke simpul sebagai berikut.



Gambar 2. 9 Pohon Berakar dengan Nilai tiap Simpul

Sehingga $\gamma(t) = 1.1.1.1.1.1.4.4.9 = 144$.

II.11 Kinerja Proses Paralel

Dalam membandingkan kinerja antara sistem multiprosesor dan sistem prosesor tunggal, maka faktor kecepatan (*speedup*) dan efisiensi yang dilambangkan dengan S_p dan E_p masing-masing harus digunakan.

S_p didefinisikan sebagai

$$S_p = \frac{t_s}{t_p} \quad (2.15)$$

di mana t_s adalah waktu eksekusi dari algoritma sekuensial terbaik yang berjalan pada satu prosesor sedangkan t_p adalah waktu eksekusi untuk memecahkan masalah yang sama pada multiprosesor. Kecepatan maksimum biasanya adalah P dengan prosesor P , yang disebut sebagai kecepatan linier. Namun, beberapa faktor akan muncul sebagai *overhead*

dalam versi paralel dan membatasi kecepatan sehingga jarang dalam mencapai hasil linier. Di antara *overheadnya* adalah :

1. Periode ketika tidak semua prosesor dapat melakukan pekerjaan yang berguna dan mengganggu.
2. Komputasi tambahan dalam versi paralel tidak muncul di versi sekuensial.
3. Waktu komunikasi antar proses.

Karena faktor-faktor inilah, kecepatan seringkali kurang dari P .

Efisiensi E_p didefinisikan sebagai

$$E_p = \frac{S_p}{P} \quad (2.16)$$

yang merupakan perbandingan kecepatan hingga jumlah prosesor. Ini digunakan untuk mengukur pemanfaatan prosesor. Jika diberikan sebagai persentase, efisiensi 100% terjadi ketika semua prosesor digunakan pada komputasi sepanjang waktu S_p sama dengan P .