

## BAB VI

### ANALISIS KINERJA PERANGKAT LUNAK

#### 6.1. Analisis Proses

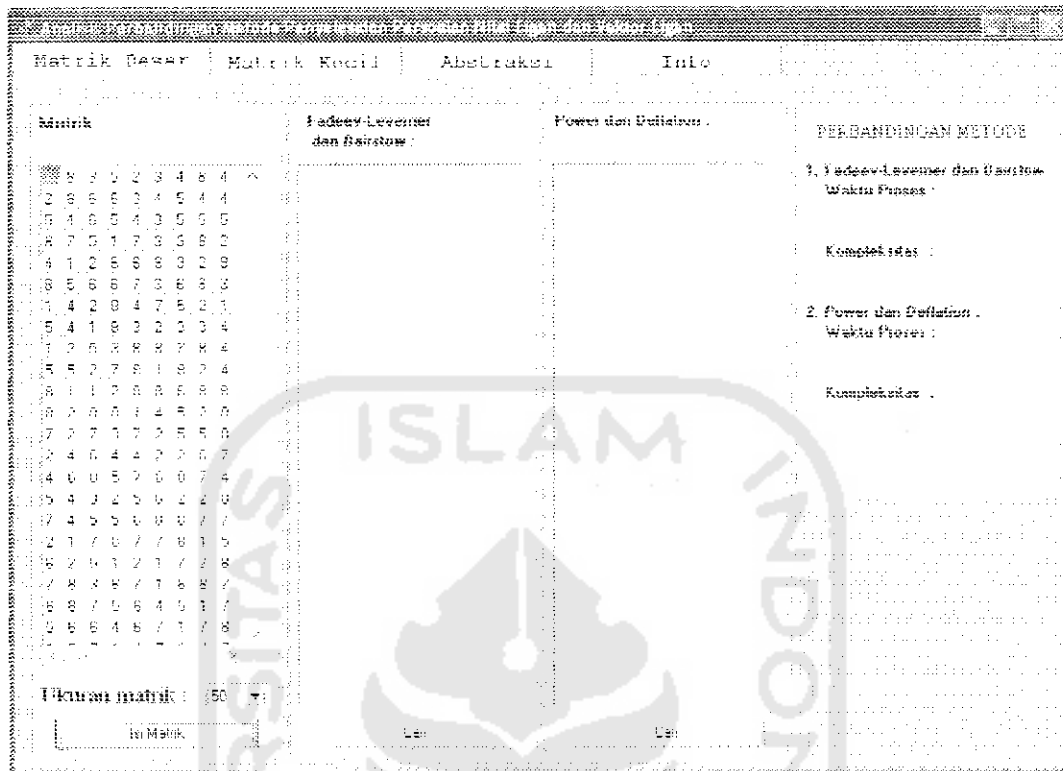
##### 6.1.1. Proses Penginputan Data

Proses ini dijalankan di dalam dua menu, yaitu menu matrik besar dan menu matrik kecil. Proses penginputan data ini sangat diperlukan oleh sistem, karena untuk mencari solusi dari suatu masalah diperlukan sejumlah data yang dapat dimanipulasi.

##### 6.1.1.1. Penginputan Data untuk Menu Matrik Besar

Pada menu ini, data yang diinputkan berupa bilangan random. Bilangan random yang dipanggil tentu tidak diketahui berapa angka yang muncul, tetapi dapat ditentukan batas awal dan batas akhir dari bilangan random itu sendiri. Sehingga, data yang akan muncul tentu berkisar antara batasan yang telah ditetapkan. Oleh karena itu, untuk memudahkan sistem dalam pencarian solusi, data yang akan ditampilkan berkisar antara 1 sampai dengan 9.

Untuk lebih jelasnya, berikut akan diperlihatkan tampilan penginputan data untuk menu matrik besar. Sebagai contoh, data yang diinputkan berupa jenis matrik biasa dengan ordo 50. Kemudian data akan ditampilkan pada *Panel Matrik*.

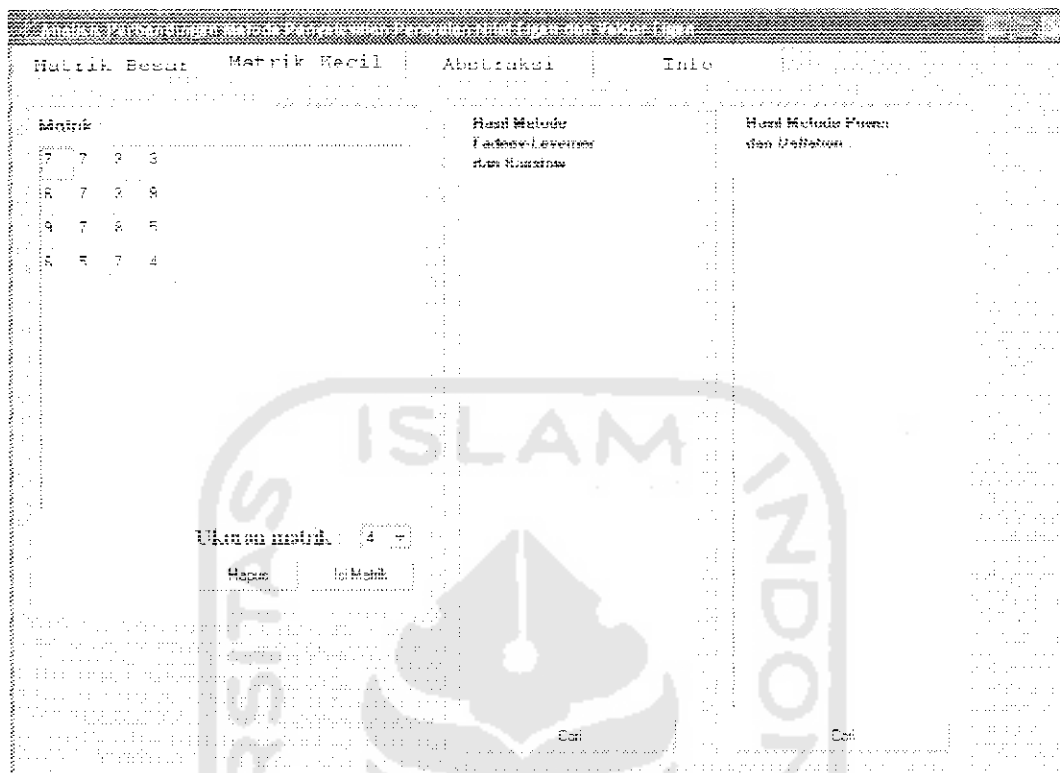


Gambar 6.1 Tampilan Penginputan Data untuk Menu Matrik Besar

#### 6.1.1.2. Penginputan Data untuk Menu Matrik Kecil

Pada menu ini, data yang diinputkan berupa bilangan random, maupun bilangan yang dapat diisikan langsung oleh pengguna. Untuk itu, pengguna dapat secara aktif menginputkan data sesukanya dengan ketentuan bahwa matrik berkisar antara ordo 3 sampai dengan ordo 10.

Untuk lebih jelasnya, berikut akan diperlihatkan tampilan penginputan data untuk menu matrik kecil. Sebagai contoh, data yang diinputkan berordo 4.



*Gambar 6.2 Tampilan Penginputan Data untuk Menu Matrik Kecil*

### 6.1.2. Proses Pemecahan Solusi

Proses ini dijalankan di dalam dua menu, yaitu menu matrik besar dan menu matrik kecil. Proses pemecahan solusi adalah inti daripada sistem itu sendiri, karena untuk mencari solusi dari suatu masalah diperlukan suatu proses yang tepat dan akurat dalam mencari hasil. Proses pemecahan solusi baru akan berjalan, jika *Tombol Cari* pada menu ditekan.

Dalam proses ini, perangkat lunak yang dibangun tentu tidak lepas dari suatu kesalahan. Kesalahan yang kadang terjadi ini diakibatkan dari kesalahan logika, yaitu apabila sistem dalam kondisi yang tidak baik. Artinya, data-data yang ada dapat menimbulkan pembagian dengan nol dan galat pembulatan.

### 6.1.3. Proses Penampilan Hasil

Proses ini dijalankan di dalam dua menu, yaitu menu matrik besar dan menu matrik kecil. Proses penampilan hasil ini sangatlah diperlukan oleh sistem, karena dapat mengetahui hasil dari solusi masalah yang sedang dihadapi.

#### 6.1.3.1. Penampilan Hasil untuk Menu Matrik Besar

Pada menu ini, hasil yang ditampilkan berupa hasil dari perhitungan keempat metode dan hasil perbandingan dari keempat metode, yang terdiri dari waktu eksekusi dan kompleksitas

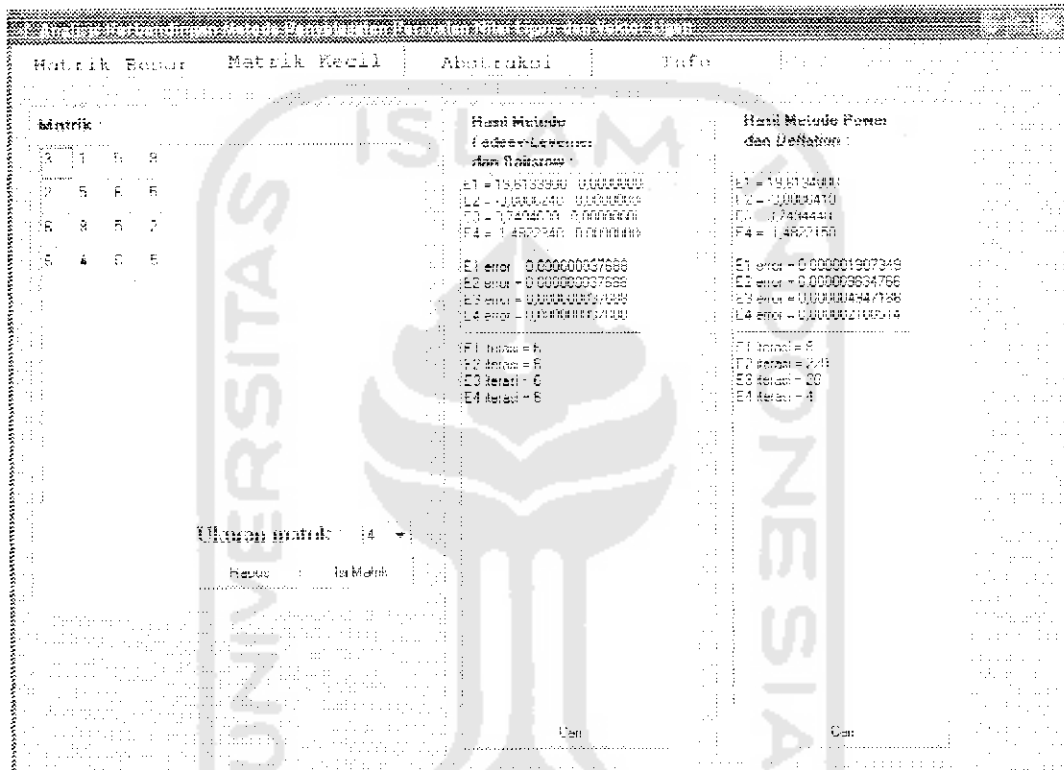
Untuk lebih jelasnya, berikut akan diperlihatkan tampilan hasil yang diperoleh dari contoh data yang telah ada.

Matrik Besar	Matrik Kecil	Abstraksi	Info
Matrik : 0 3 5 2 7 4 8 4 2 8 6 6 3 4 5 4 4 3 4 6 5 4 3 5 5 5 6 7 5 7 7 3 3 8 2 4 1 2 6 6 8 3 2 8 8 0 8 8 7 3 8 8 3 1 4 2 8 4 7 0 2 1 5 4 1 8 3 2 3 3 4 1 2 0 8 8 8 7 8 4 5 5 2 7 0 1 0 2 4 8 1 1 2 8 8 8 8 8 8 2 8 8 1 4 5 2 8 7 2 7 3 7 2 5 5 8 2 4 8 4 4 2 2 6 7 4 5 5 5 7 5 8 7 4 5 4 3 2 6 6 2 2 8 7 4 5 5 6 8 8 7 7 5 1 7 8 7 7 8 1 5 5 2 5 1 2 1 7 2 8 7 8 3 3 7 1 0 8 7 6 8 3 5 6 4 2 1 7 5 0 6 4 6 7 1 7 0	Listensi matrik dan Raitrow : E1 = 229.95000 0.90000 E2 = 94.87000 0.88000 E3 = 171.70000 0.70700 E4 = 171.70000 0.70700 E5 = 15.11000 170.00000 E6 = 15.11000 170.00000 E7 = 131.34000 112.71000 E8 = 131.34000 112.71000 E9 = 178.10000 12.78000 E10 = 178.10000 12.78000 E11 = 235.00000 12.00000 E12 = 235.00000 12.00000 E13 = 164.12000 54.98000 E14 = 164.12000 54.98000 E15 = 172.87000 11.21700 E16 = 172.87000 11.21700 E17 = 79.09000 104.08000 E18 = 79.09000 104.08000 E19 = 152.41000 81.00000 E20 = 152.41000 81.00000 E21 = 127.27000 118.47000 E22 = 127.27000 118.47000 E23 = 92.32000 147.05000 E24 = 92.32000 147.05000 E25 = 51.44500 185.70000 E26 = 51.44500 185.70000 E27 = 99.24500 147.59000 E28 = 99.24500 147.59000 E29 = 168.90000 25.42000 E30 = 168.90000 25.42000 E31 = 144.67000 84.90100 E32 = 144.67000 84.90100 E33 = 27.20000 102.00000	Proses dan Definisi : F1 = 229.95000 F2 = 94.87000 F3 = 171.70000 F4 = 171.70000 F5 = 15.11000 F6 = 15.11000 F7 = 131.34000 F8 = 131.34000 F9 = 178.10000 F10 = 178.10000 F11 = 235.00000 F12 = 235.00000 F13 = 164.12000 F14 = 164.12000 F15 = 172.87000 F16 = 172.87000 F17 = 79.09000 F18 = 79.09000 F19 = 152.41000 F20 = 152.41000 F21 = 127.27000 F22 = 127.27000 F23 = 92.32000 F24 = 92.32000 F25 = 51.44500 F26 = 51.44500 F27 = 99.24500 F28 = 99.24500 F29 = 168.90000 F30 = 168.90000 F31 = 144.67000 F32 = 144.67000 F33 = 27.20000	PERBANDINGAN METODE : 1. Metode Gauss dan Raitrow Waktu Proses : 172 mtk Kompleksitas : 10.074.700 Langkah 2. Metode Gauss Waktu Proses : 1 mtk Kompleksitas : 62.002.000 Langkah

Gambar 6.3 Tampilan Hasil Perhitungan untuk Menu Matrik Besar

### 6.1.3.2. Penampilan Hasil untuk Menu Matrik Kecil

Pada menu ini, hasil yang ditampilkan berupa hasil perhitungan dari keempat metode saja. Untuk lebih jelasnya, berikut ini akan diperlihatkan tampilan hasil yang diperoleh dari contoh data yang telah ada.



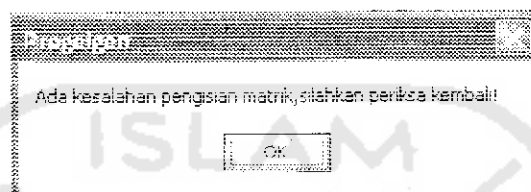
Gambar 6.4 Tampilan hasil perhitungan untuk menu matrik kecil

### 6.1.4. Analisis Proses dan Pengujian Tidak Normal

Analisis proses dan pengujian tidak normal dilakukan dengan memberikan masukan yang menurut spesifikasi dalam pembuatan program tidak diijinkan. Pengujian dilakukan untuk memeriksa respon perangkat lunak terhadap hal yang tidak diijinkan program tersebut.

#### 6.1.4.1. Analisis Proses dan Pengujian pada Menu Matrik Kecil dan Besar

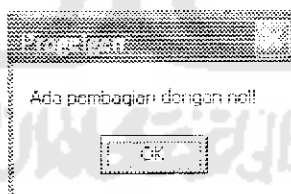
Pada menu ini, apa bila pengguna salah memasukkan data pada matrik kecil, atau ada data yang dikosongkan, maka akan muncul pesan kesalahan seperti pada gambar 6.5 berikut.



Gambar 6.5 Tampilan Informasi untuk kesalahan pengisian data matrik kecil

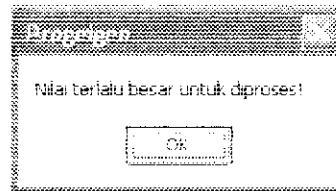
Dalam proses ini, perangkat lunak yang dibuat tentu tidak lepas dari suatu kesalahan. Kesalahan yang dapat timbul antara lain adanya pembagian dengan nol, nilai yang diolah terlalu besar, ataupun adanya kesalahan perhitungan.

Apabila terjadi suatu pembagian dengan nol, maka akan muncul pesan kesalahan seperti berikut :



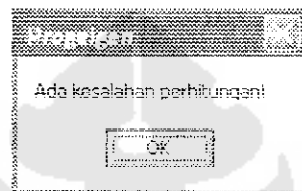
Gambar 6.6 Tampilan pesan untuk pembagian dengan nol

Setiap perangkat lunak tentu mempunyai keterbatasan untuk besarnya nilai yang diolah, apabila nilai yang diolah terlalu besar untuk diproses, maka akan muncul pesan kesalahan seperti berikut :



Gambar 6.7 Tampilan pesan untuk nilai yang terlalu besar

Dan apabila terjadi kesalahan logika pada program, sehingga menimbulkan kesalahan perhitungan, maka akan muncul pesan kesalahan seperti berikut :



Gambar 6.8 Tampilan pesan untuk kesalahan perhitungan

## 6.2. Analisis terhadap Antarmuka Perangkat Lunak

Antarmuka menjadi hal yang sangat penting dalam suatu perangkat lunak. Perangkat lunak yang mempunyai (*interface*) yang baik harus bersifat ramah pengguna (*user friendly*), artinya mudah digunakan tanpa melalui proses belajar yang rumit dan tidak membingungkan pengguna.

Antarmuka yang digunakan pada perangkat lunak ini dibuat semaksimal mungkin bersifat *user friendly*. Hal ini dapat dilihat dari beberapa segi berikut ini.

### 6.2.1. Level Keahlian Pemakai

Perangkat lunak ini dikembangkan untuk memenuhi kebutuhan para pengguna yang bersifat *Unknowledgeable User*, artinya pengguna tidak perlu mengetahui fungsi dan prosedur yang ada dan alur kerja sistem dari perangkat lunak

### 6.2.2. Interaksi Manusia dan Komputer

Perangkat lunak ini cukup bersifat *komunikatif*, artinya mudah dimengerti oleh pengguna. Jika terjadi kesalahan dalam menjalankan proses program, maka sistem akan memberikan tanggapan kepada pengguna. Perangkat lunak ini juga bersifat *interaktif*, artinya dapat melakukan interaksi dengan pengguna.

### 6.2.3. Input Data dan Proses

Penginputan data maupun proses yang lebih sedikit serta pembatasan input atau proses (semua objek dibatasi penggunaannya baik dengan penguncian *keyboard* ataupun dengan objek-objek yang bersifat optional seperti *ComboBox*) menjadi perhatian dalam perancangan perangkat lunak ini. Hal ini diharapkan dapat mengurangi kesalahan oleh *user*.



### 6.3. Analisis Kinerja dan Pengujian Perangkat Lunak

Kinerja yang dihasilkan oleh perangkat lunak ini dapat dilihat dari beberapa sisi, yaitu :

#### 6.3.1. Analisis Kinerja untuk Data Masukan (*Input*)

Dalam menganalisa perbandingan antar keempat metode yang digunakan, perangkat lunak ini dapat menerima masukan (input) sampai dengan 1 juta data atau dengan kata lain matriks yang digunakan sampai berordo 1000 x 1000, bahkan lebih dari itu. Hal ini dikarenakan dalam prosedur pemasukan data menggunakan tipe data *pointer*, yang besarnya tergantung dari ukuran memori komputer.

#### 6.3.2. Analisis Kinerja Pemrosesan

Dalam melakukan suatu proses, perangkat lunak ini selalu membandingkan keempat metode yang digunakan. Untuk lebih jelasnya, berikut akan dibahas karakteristik dari keempat metode dalam melakukan penyelesaian sistem persamaan linier.

### 6.3.2.1. Analisis Kinerja Pemrosesan dengan Metode Fadeev-Leverrier dan Bairstow

Dalam melakukan penyelesaian sistem persamaan homogen, metode ini menempati urutan kedua dalam melakukan proses. Ini terbukti dari fungsi langkah yang dapat dirumuskan sebagai berikut :  $n^4 + n^3 + 2n^2 - 2n$ .

**Big-O** operasi pada fungsi di atas adalah sebagai berikut :

1. Pencarian hasil sebanyak  $n$  mempunyai  $O(n^4)$ .
2. Pengambilan data pada elemen tertentu yang diketahui posisinya mempunyai  $O(n)$ , karena memerlukan waktu yang bergantung dari jumlah elemen yang ada.

### 6.3.2.2. Analisis Kinerja Pemrosesan dengan Metode Power dan Deflation

Dalam melakukan penyelesaian sistem persamaan homogen, metode ini bisa dikatakan lebih cepat dari metode Fadeev-Leverrier dan Bairstow dalam melakukan proses. Ini terbukti dari fungsi langkah yang dapat dirumuskan sebagai berikut :  $(4 + C)n^3 + (2C + 3)n^2$ . Konstanta yang ada tergantung dari jumlah iterasi yang telah ditetapkan

**Big-O** operasi pada fungsi di atas adalah sebagai berikut :

1. Pencarian hasil sebanyak  $n$  mempunyai  $O(n^3)$ .
2. Pengambilan data pada elemen tertentu yang diketahui posisinya mempunyai  $O(n)$ , karena memerlukan waktu yang bergantung dari jumlah elemen yang ada.

### 6.3.3. Pengujian Perangkat Lunak

Sebelum perangkat lunak ini diterapkan, maka program harus bebas terlebih dahulu dari berbagai kesalahan. Pengujian perangkat lunak dilakukan untuk menemukan kesalahan yang mungkin terjadi, selain itu pengujian perangkat lunak dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan kebutuhan dari perancangan.

Pengujian ini adalah untuk jenis kesalahan dari sistem berupa kesalahan logika (*logical error*). Kesalahan logika adalah kesalahan dari logika program yang telah dibuat. Kesalahan seperti ini sulit ditemukan, dikarenakan tidak adanya pemberitahuan mengenai kesalahannya dan tetap akan didapatkan hasil walaupun hasil sebenarnya adalah salah.

Pengujian sistem (*system testing*) dilakukan untuk memeriksa serta menilai kekonvergenan antara data pengujian dengan jalannya sistem yang diimplementasikan. Tujuan utama dari pengujian sistem ini adalah untuk mengetahui sampai sejauh mana sistem memberikan hasil yang akurat dengan pemasukan data yang ada.

Untuk melihat apakah kinerja dari perangkat lunak sesuai yang diharapkan, maka diperlukan suatu pengujian terhadap perangkat lunak ini. Data yang digunakan untuk pengujian ialah data yang diambil secara random oleh komputer. Dari hasil pengujian didapatkan waktu dan kompleksitas yang dibutuhkan dalam penyelesaian sistem persamaan homogen untuk setiap metode.

Solusi yang telah didapatkan dari masing-masing metode kemudian dibandingkan untuk mencari metode mana yang terbaik, yaitu yang membutuhkan waktu sedikit dalam penyelesaian sistem persamaan homogen.

### 6.3.3.1. Pengujian Perangkat Lunak Untuk Menu Matrik Besar

Untuk hasil pengujian pada menu matrik besar, dapat dilihat pada tabel berikut ini :

Ukuran Matrik	Jenis Metode	
	Fadeev-Leverrier dan Bairstow	Power dan Deflation
50	187 milidetik	937 milidetik
100	3 detik 204 milidetik	7 detik 422 milidetik
150	16 detik 813 milidetik	23 detik 406 milidetik
200	1 menit 13 detik 375 milidetik	2 menit 27 detik 516 milidetik
250	3 menit 12 detik 891 milidetik	4 menit 51 detik 485 milidetik
300	6 menit 25 detik 375 milidetik	9 menit 40 detik 438 milidetik

*Tabel 6.1 Hasil Pengujian Sistem Berdasarkan Waktu Eksekusi*

Ukuran Matrik	Jenis Metode	
	Fadeev-Leverrier dan Bairstow	Power dan Deflation
50	6,379,900	130,507,500
100	101,019,800	1,024,030,000
150	509,669,700	3,433,567,500
200	1,608,079,600	8,112,120,000
250	3,921,999,500	15,812,687,500
300	8,127,179,400	27,288,270,000

*Tabel 6.2 Hasil Pengujian Sistem Berdasarkan Kompleksitas Algoritma*

Pada tabel 6.2 di atas, dapat dibuat suatu perbandingan antara hasil perhitungan dengan hasil pengujian sistem. Untuk lebih jelasnya, berikut akan diterangkan perhitungan kompleksitas pada masing-masing metode.

#### 1. Metode Fadeev-Leverrier

Dari tabel 6.2 di atas, anggap nilai basis terletak pada ukuran matriks ordo 50. Untuk ukuran matriks ordo 300, berarti data yang ada sebesar 6 kalinya. Apabila untuk ukuran matriks ordo 50 kompleksitasnya adalah 6,379,900 langkah, berarti untuk ukuran matriks ordo 300 kompleksitasnya adalah  $(6)^4 \times 6,379,900 = 8,268,350,400$  langkah.

#### 2. Metode Power dan Deflation

Dari tabel 6.2 di atas, anggap nilai basis terletak pada ukuran matriks ordo 50. Untuk ukuran matriks ordo 300, berarti data yang ada sebesar 6 kalinya. Apabila untuk ukuran matriks ordo 50 kompleksitasnya adalah 1 010 000 langkah, berarti untuk ukuran matriks ordo 300 kompleksitasnya adalah  $(6)^3 \times 130,507,500 = 28,189,620,000$  langkah.

Dari hasil yang diperoleh, terbukti bahwa hasil perhitungan mendekati hasil pengujian sistem. Ini berarti kompleksitas yang digunakan sesuai dengan fungsi langkah yang diterapkan pada masing-masing metode.

### 6.3.3.2. Pengujian Perangkat Lunak Untuk Menu Matrik Kecil

Pada pengujian untuk menu matrik kecil, dapat dilihat hasil program pada gambar-gambar berikut :

Matrik :	Hasil Metode Faktor-Leverier dari Kaulstow :	Hasil Metode Power dan Deflation :
<pre> 4  0  7  9  2 5  4  2  7  1 4  3  2  3  8 3  5  6  9  4 1  3  6  1  6           </pre>	<pre> E1 = 32.8792900  0.0000000 E2 = 6.1111560  0.0000000 E3 = -3.4727360  0.0000000 E4 = 2.0663950  0.0000000 E5 = -1.7376100  0.0000000  F1 error = 0.00000005905 F2 error = 0.00000002590 F3 error = 0.00000002305 F4 error = 0.00000003465 F5 error = 0.00000003405  E1 iterasi = 5 E2 iterasi = 5 E3 iterasi = 5 E4 iterasi = 5 E5 iterasi = 5           </pre>	<pre> E1 = 32.8792900 E2 = 6.1111560 E3 = -3.4727360 E4 = 2.0663950 E5 = -1.7376100  F1 error = 0.00000005905 F2 error = 0.00000002590 F3 error = 0.00000002305 F4 error = 0.00000003465 F5 error = 0.00000003405  E1 iterasi = 9 E2 iterasi = 32 E3 iterasi = 10000 E4 iterasi = 10000 E5 iterasi = 10000           </pre>

Gambar 6.9 Hasil perhitungan pertama pada matrik kecil

Matrik :	Hasil Metode Faktor-Leverier dari Kaulstow :	Hasil Metode Power dan Deflation :
<pre> 2  5  9  7 4  0  4  2  0 7  9  5  5  5 0  0  4  3  4 6  6  9  8  1           </pre>	<pre> E1 = 21.6327200  0.0000000 E2 = 6.1967400  0.0000000 E3 = 2.4611210  0.0000000 E4 = -2.0000000  0.0000000 E5 = -0.0000000  0.0000000  F1 error = 0.00000005905 F2 error = 0.00000002590 F3 error = 0.00000002305 F4 error = 0.00000003465 F5 error = 0.00000003405  E1 iterasi = 22 E2 iterasi = 22 E3 iterasi = 13 E4 iterasi = 13 E5 iterasi = 13           </pre>	<pre> E1 = 21.6327200 E2 = 6.1967400 E3 = 2.4611210 E4 = -2.0000000 E5 = -0.0000000  F1 error = 0.00000005905 F2 error = 0.00000002590 F3 error = 0.00000002305 F4 error = 0.00000003465 F5 error = 0.00000003405  E1 iterasi = 12 E2 iterasi = 16 E3 iterasi = 67 E4 iterasi = 65 E5 iterasi = 10000           </pre>

Gambar 6.10 Hasil perhitungan kedua pada matrik kecil

Matrik :	Hasil Metode Faddeev-Leverrier dan Bairstow	Hasil Metode Power dan Deflation :
$\begin{bmatrix} 0 & 7 & 7 & 1 & 2 \\ 0 & 2 & 0 & 3 & 0 \\ 1 & 0 & 1 & 2 & 2 \\ 3 & 2 & 7 & 1 & 4 \\ 2 & 1 & 0 & 6 & 3 \end{bmatrix}$	E1 = 21.9911400 0.0000000 E2 = 5.8080889 0.0000000 E3 = 0.2120550 0.0000000 E4 = 2.0417140 0.0000000 E5 = -0.0363334 0.0000000 F1 error = 0.00000004841 F2 error = 0.000000004941 F3 error = 0.00000001911 F4 error = 0.000000016676 F5 error = 0.000000010076 F1 iterasi = 5 F2 iterasi = 5 F3 iterasi = 5 F4 iterasi = 7 F5 iterasi = 7	E1 = 21.9911439 E2 = 5.8080889 E3 = 0.2120550 E4 = 2.0417200 E5 = -0.0363407 F1 error = 0.00000059675 F2 error = 0.00000059605 F3 error = 0.00000067055 F4 error = 0.00000089407 F5 error = 0.0000017344052 F1 iterasi = 13 F2 iterasi = 41 F3 iterasi = 17 F4 iterasi = 21 F5 iterasi = 10000

Gambar 6.11 Hasil perhitungan ketiga pada matrik kecil

Dari gambar-gambar di atas, dapat diterangkan sebagai berikut :

1. Maksimum iterasi untuk kedua metode ditetapkan maksimum 10,000 iterasi.
2. Nilai toleransi kesalahan untuk kedua metode diatas ditetapkan 0.00000001
3. Nilai eigen yang dihasilkan tidak ada yang imajiner.

Berdasarkan hasil yang terlihat pada gambar-gambar di atas, dapat dilihat bahwa metode Faddeev-Leverrier dan Bairstow membutuhkan iterasi yang lebih sedikit untuk mendapatkan nilai eigen dengan toleransi kesalahan yang telah ditetapkan.

Pada gambar-gambar berikut, dapat dilihat hasil perhitungan pada menu matrik kecil dengan nilai eigen yang imajiner :

Matrik :	Hasil Metode Faddeev-Levina dan Rutisow :	Hasil Metode Power dan Deflation :
<pre> 7  2  7  6  4 2  2  4  7  3 6  2  7  7  6 4  6  6  8  2 3  3  3  1  9                     </pre>	<pre> E1 = 24.0000000 0.0000000 E2 = 5.5000000 0.0000000 E3 = 2.1673600 0.0000000 E4 = 1.1520000 1.0000000 E5 = 2.1520000 -1.5000000  E1 error = 0.000000016764 E2 error = 0.000000010302 E3 error = 0.000000016302 E4 error = 0.000000016764 E5 error = 0.000000016764  E1 iterasi = 8 E2 iterasi = 19 E3 iterasi = 19 E4 iterasi = 8 E5 iterasi = 8                     </pre>	<pre> E1 = 24.0000000 E2 = 5.5000000 E3 = 2.1673600 E4 = 7.6279460 E5 = 2.1070000  E1 error = 0.000000059606 E2 error = 0.000000074506 E3 error = 0.000000025500 E4 error = 0.000000036447 E5 error = 0.000000003407  E1 iterasi = 11 E2 iterasi = 21 E3 iterasi = 111111 E4 iterasi = 50000 E5 iterasi = 162                     </pre>

Gambar 6.12 Hasil perhitungan imajiner pertama pada matrik kecil

Matrik :	Hasil Metode Faddeev-Levina dan Rutisow :	Hasil Metode Power dan Deflation :
<pre> 7  7  2  1  2 4  6  6  8  9 6  9  3  3  9 3  8  9  7  9 3  9  3  3  7                     </pre>	<pre> E1 = 27.2768000 0.0000000 E2 = 5.0407000 0.0000000 E3 = 5.5849600 0.0000000 E4 = 0.0384984 0.6687331 E5 = 0.0384984 -0.6687331  E1 error = 0.000000022666 E2 error = 0.000000022666 E3 error = 0.000000022666 E4 error = 0.000000020000 E5 error = 0.000000026308  E1 iterasi = 4 E2 iterasi = 4 E3 iterasi = 4 E4 iterasi = 4 E5 iterasi = 4                     </pre>	<pre> E1 = 27.2768000 E2 = 5.0407000 E3 = 5.5849600 E4 = 0.1741519 E5 = 0.7733022  E1 error = 0.000000005005 E2 error = 0.000000078804 E3 error = 0.000000088006 E4 error = 0.541004527000 E5 error = 0.000000000000  E1 iterasi = 11 E2 iterasi = 50000 E3 iterasi = 10 E4 iterasi = 00000 E5 iterasi = 8975                     </pre>

Gambar 6.13 Hasil perhitungan imajiner kedua pada matrik kecil

Matrik :	Hasil Metode Faddeev-Levina dan Rutisow :	Hasil Metode Power dan Deflation :
<pre> 7  9  1  7  2 0  4  3  1  2 4  8  8  1  6 5  2  5  2  6 6  4  1  8  7                     </pre>	<pre> E1 = 27.7749000 0.0000000 E2 = 4.3368000 0.5671572 E3 = 4.3368000 -0.5671572 E4 = 4.2144600 0.0000000 E5 = 0.7662000 0.0000000  E1 error = 0.000000016837 E2 error = 0.000000010537 E3 error = 0.000000010537 E4 error = 0.000000022455 E5 error = 0.000000022455  E1 iterasi = 6 E2 iterasi = 6 E3 iterasi = 6 E4 iterasi = 55 E5 iterasi = 55                     </pre>	<pre> E1 = 27.7749000 E2 = 3.9020000 E3 = 4.0870200 E4 = 4.2144600 E5 = 0.7056642  E1 error = 0.000000056005 E2 error = 0.051370200040 E3 error = 0.000000029602 E4 error = 0.000000050005 E5 error = 0.000000076744  E1 iterasi = 11 E2 iterasi = 111111 E3 iterasi = 177 E4 iterasi = 37 E5 iterasi = 14000                     </pre>

Gambar 6.14 Hasil perhitungan imajiner ketiga pada matrik kecil



Dari gambar-gambar di atas, dapat diterangkan sebagai berikut

1. Maksimum iterasi untuk kedua metode ditetapkan maksimum 50,000 iterasi.
2. Nilai toleransi kesalahan untuk kedua metode diatas ditetapkan 0,0000001.
3. Nilai eigen yang dihasilkan ada yang imajiner.

Berdasarkan hasil yang terlihat pada gambar-gambar di atas, dapat dilihat bahwa metode Power dan Deflation menghasilkan nilai toleransi kesalahan yang cukup besar untuk nilai eigen yang imajiner atau dengan kata lain, nilai eigen yang memiliki nilai hampir sama dengan nilai eigen berikutnya. Hal ini dapat dilihat pada gambar 6.13 pada nilai eigen ketiga mencapai nilai kesalahan 0,5410845277969.

