



Perhitungan Kompleksitas Metode *Golden Section* dalam Optimasi Parameter Pemulusan Eksponensial Ganda Brown dan Holt

Nurrahim Dwi Saputra¹, Abdul Aziz², Bambang Harjito³

Jurnal Algoritma
Universitas Sebelas Maret
Jl. Ir. Sutami No.36, Kentingan, Kec. Jebres, Kota Surakarta, Jawa Tengah 57126 Indonesia
E-mail: campus@mail.uns.ac.id

¹nurrahim@student.uns.ac.id

²aaziz@staff.uns.ac.id

³bambang_harjito@staff.uns.ac.id

Abstrak – Algoritma yang baik, selain memiliki nilai galat yang minimum, diharapkan juga memiliki waktu eksekusi dan ruang memori yang efisien. Penelitian ini melakukan perhitungan kompleksitas waktu menggunakan pendekatan teoretis dan eksperimental dari metode *golden section* yang digunakan sebagai optimator parameter dalam prediksi menggunakan pemulusan eksponensial ganda Brown dan Holt. Hasil menunjukkan bahwa metode *golden section* untuk optimasi parameter α pada pemulusan eksponensial ganda Brown, metode *modified golden section* untuk optimasi parameter α dan γ pada pemulusan eksponensial ganda Holt, metode pemulusan eksponensial ganda Brown, dan metode pemulusan eksponensial ganda Holt memiliki kompleksitas waktu linear $O(n)$, dengan n adalah banyaknya data.

Kata Kunci – *Golden Section*; Kompleksitas Waktu; Optimasi Parameter; Pemulusan Eksponensial Ganda Brown; Pemulusan Eksponensial Ganda Holt.

I. PENDAHULUAN

Sebuah algoritma dapat dikatakan baik jika memiliki nilai galat yang minimum serta diharapkan memiliki waktu eksekusi dan ruang memori yang efisien. Metode optimasi *golden section* telah diterapkan untuk optimasi pemulusan ganda Brown dan Holt dengan hasil galat (*mean absolute percentage error/MAPE*) yang kecil ($10\% < MAPE \leq 20\%$) sehingga dapat dikategorikan sebagai estimator dan prediktor yang baik [1]. Namun, selain memiliki galat yang minimal, metode yang bagus adalah yang efisien dalam jumlah waktu dan ruang memori yang dibutuhkannya ketika diimplementasikan dalam algoritma program. Penelitian sebelumnya [1] hanya melakukan implementasi metode, namun belum dilakukan pengujian uji kompleksitas terhadapnya. Penelitian ini melakukan perhitungan kompleksitas waktu eksekusi menggunakan pendekatan teoretis dan eksperimental dari metode *golden section* yang digunakan sebagai optimator parameter dalam prediksi menggunakan pemulusan eksponensial ganda Brown dan Holt.

Secara teoretis, model abstrak pengukuran waktu/ruang harus independen terhadap mesin yang digunakan untuk kompilasi. Model abstrak tersebut dapat digunakan untuk membandingkan beberapa algoritma. Model abstrak seperti ini disebut kompleksitas algoritma [2]. Terdapat dua macam kompleksitas algoritma, yakni kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu dapat dikatakan merupakan jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi ukuran input n . Kompleksitas ruang dapat dikatakan merupakan jumlah memori yang digunakan oleh struktur data yang terdapat di dalam algoritma sebagai fungsi dari ukuran input n . Kompleksitas waktu biasanya direpresentasikan dalam bentuk kompleksitas waktu asimtotik. Tabel 1 menunjukkan kelompok algoritma berdasarkan kompleksitas waktu asimtotiknya [3].

Tabel 1: Kelompok Algoritma Berdasarkan Kompleksitas Waktu Asimtotiknya

| Kelompok Algoritma | Nama |
|--------------------|--------------|
| $O(1)$ | Konstan |
| $O(\log n)$ | Logaritmik |
| $O(n)$ | Linear |
| $O(n \log n)$ | $n \log n$ |
| $O(n^2)$ | Kuadratik |
| $O(n^3)$ | Kubik |
| $O(2n)$ | Eksponensial |
| $O(n!)$ | Faktorial |

Pemulusan eksponensial adalah metode yang umum digunakan untuk melakukan prediksi. Beberapa penelitian telah melakukan prediksi berbagai jenis data menggunakan metode pemulusan eksponensial [1], [4]–[8]. Pemulusan eksponensial (*exponential smoothing*) adalah suatu prosedur yang mengulang perhitungan secara terus-menerus dengan menggunakan data terbaru yang didasarkan pada perhitungan rata-rata (pemulusan) data-data masa lalu secara eksponensial serta setiap data diberi bobot dengan ketentuan data yang lebih baru diberi bobot yang lebih besar [9]. Bentuk umum metode ini ditunjukkan pada persamaan (1) [10].

$$F_{t+1} = \alpha X_t + (1 - \alpha)F_t \quad (1)$$

dengan

F_{t+1} = prediksi satu periode ke depan

X_t = data aktual periode t

F_t = prediksi pada periode t

α = parameter pemulusan ($0 < \alpha < 1$)

Implikasi pemulusan eksponensial dapat dilihat dengan lebih baik bila persamaan (1) diperluas dengan substitusi F_t dengan komponennya. Jika proses substitusi diulang dengan cara mengganti F_{t-1} , F_{t-2} , dan seterusnya dengan komponennya, diperoleh bahwa bobot untuk X_t , X_{t-1} , dan seterusnya menurun secara eksponensial, sehingga metode ini disebut pemulusan eksponensial [10].

Salah satu permasalahan dalam metode pemulusan eksponensial adalah penentuan nilai parameter pemulusan sehingga diperoleh galat yang minimum. Optimasi parameter pemulusan ini dapat dilakukan menggunakan metode optimasi nonlinear [10]. Salah satu metode optimasi nonlinear adalah *golden section* [11]. Pada umumnya, metode *golden section* digunakan untuk menyelesaikan NLP (*non-linear programming*) satu variabel yang berbentuk

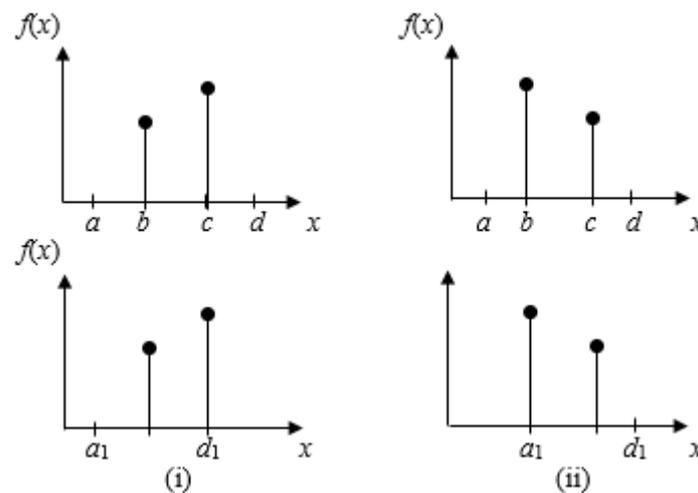
$$\begin{aligned} \text{maksimasi atau minimasi} & : f(x) \\ \text{dengan kendala} & : a \leq x \leq d \quad [2] \end{aligned}$$

Metode *golden section* menggunakan prinsip mengurangi daerah batas x yang mungkin menghasilkan nilai fungsi objektif optimum (maksimum atau minimum) secara iteratif. Misalkan pada suatu tahap iterasi nilai fungsi optimum mungkin terletak pada interval $x[a,d]$. Kemudian menentukan dua nilai x yang simetris dalam interval tersebut yaitu b dan c , dan interval kemungkinan fungsi berharga optimum dikurangi dari $[a,d]$ menjadi $[a,c]$ atau $[b,d]$ tergantung dari nilai di $x = b$ dan di $x = c$ [5]. Ilustrasi tersaji pada Gambar 1.

Pada setiap tahapan iterasi ditentukan dua buah titik di dalam interval yang ada. Namun, untuk tujuan penghematan langkah perhitungan, pada setiap tahapan iterasi hanya ditentukan sebuah titik baru. Titik yang lain adalah titik yang ditentukan pada tahap sebelumnya. Misalnya, interval telah dapat dikurangi dari $[a,d]$ menjadi $[a,c]$. Interval $[a,c]$ merupakan interval yang baru sehingga dapat dituliskan menjadi $[a_1,d_1]$. Hanya ditentukan satu titik baru yaitu b_1 karena titik b dijadikan titik c_1 , sehingga diperoleh hubungan seperti pada persamaan berikut.

$$b_1 = ra_1 + (1 - r)d_1 \quad (2)$$

$$c_1 = a_1 + d_1 - b_1 \quad (3)$$



Gambar 1: Proses Pengurangan Interval Pencarian x Optimum (i) dari $[a,d]$ Menjadi $[a,c]$ (ii) dari $[a,d]$ Menjadi $[b,d]$ [5]

Nilai r adalah $0,5(-1 - \sqrt{5})$ atau $0,5(-1 + \sqrt{5})$. Agar didapat interval yang semakin kecil diperlukan syarat yakni $0 < r < 1$ sehingga nilai r yang digunakan adalah $0,5(-1 + \sqrt{5})$ atau sekitar 0,61803.

Metode *golden section* yang hanya dapat menyelesaikan NLP dengan satu variabel dapat dimodifikasi untuk menyelesaikan permasalahan dengan banyak variabel [4]. Bentuk umum NLP tersebut adalah

maksimasi atau minimasi : $f(x_1, x_2, x_3, \dots, x_n)$

dengan kendala : $\begin{cases} a_1 \leq x \leq d_1 \\ a_2 \leq x \leq d_2 \\ a_3 \leq x \leq d_3 \\ \vdots \\ a_n \leq x \leq d_n \end{cases}$

Metode *golden section* telah diterapkan untuk optimasi parameter pemulusan eksponensial, seperti pada metode *single exponential smoothing* (SES), *double exponential smoothing* (DES) Brown dan Holt, dan *triple exponential smoothing* (TES) Holt-Winters aditif dan multiplikatif [1], [4], [5]. Penelitian ini berfokus pada metode pemulusan eksponensial ganda Brown dan Holt.

Pemulusan eksponensial ganda linear satu parameter dari Brown sesuai untuk data yang ada menunjukkan sifat tren atau dipengaruhi unsur tren. Pada metode pemulusan eksponensial ini dilakukan proses pemulusan dua kali dan pemulusan tren sebagai berikut [10].

Pemulusan eksponensial tunggal:

$$S'_t = \alpha X_t + (1 - \alpha)S'_{t-1} \tag{4}$$

Pemulusan eksponensial ganda:

$$S''_t = \alpha S'_t + (1 - \alpha)S''_{t-1} \tag{5}$$

Pemulusan tren:

$$a_t = S'_t + (S'_t - S''_t) = 2S'_t - S''_t \tag{6}$$

$$b_t = \frac{\alpha}{1 - \alpha} (S'_t - S''_t) \tag{7}$$

Ramalan:

$$F_{t+m} = a_t + b_t m \tag{8}$$

dengan

X_t = nilai aktual pada periode t

α = konstanta pemulusan ($0 < \alpha < 1$)

F_{t+m} = ramalan untuk m periode ke depan dari t

Untuk dapat menggunakan persamaan (4) dan (5), nilai S'_{t-1} dan S''_{t-1} harus tersedia. Pada $t = 1$, nilai-nilai tersebut belum tersedia dan harus ditentukan di awal periode. Hal ini dapat dilakukan dengan hanya menetapkan S'_t dan S''_t sama dengan X_t atau dengan menggunakan suatu nilai rata-rata dari beberapa nilai pertama sebagai titik awal [10].

Pemulusan eksponensial ganda dua parameter dari Holt pada dasarnya sama dengan Brown yang sesuai untuk data yang ada menunjukkan sifat tren atau dipengaruhi unsur tren. Prinsipnya sama dengan Brown, namun Holt tidak menggunakan rumus pemulusan berganda secara langsung. Sebagai gantinya, Holt memuluskan tren dengan menggunakan parameter yang berbeda dari parameter pada deret asli. Ramalan pemulusan eksponensial Holt diperoleh menggunakan dua konstanta pemulusan dan tiga persamaan [10].

Pemulusan keseluruhan:

$$S_t = \alpha X_t + (1 - \alpha)(S_{t-1} + b_{t-1}) \quad (9)$$

Pemulusan tren:

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1} \quad (10)$$

Ramalan:

$$F_{t+m} = S_t + b_t m \quad (11)$$

dengan

X_t = nilai aktual pada periode t

α = konstanta pemulusan keseluruhan ($0 < \alpha < 1$)

γ = konstanta pemulusan tren ($0 < \gamma < 1$)

F_{t+m} = ramalan untuk m periode ke depan dari t

Untuk menggunakan persamaan (9) dan (10) diperlukan proses inisialisasi melalui dua taksiran, yakni untuk S_1 dan b_1 . Untuk taksiran S_1 dapat menggunakan nilai aktual X_1 . Untuk taksiran tren terdapat beberapa kemungkinan (persamaan (12) dan (13)) berikut [4].

$$b_1 = X_2 - X_1 \quad (12)$$

$$b_1 = \frac{(X_2 - X_1) + (X_3 - X_2) + (X_4 - X_3)}{3} \quad (13)$$

II. URAIAN PENELITIAN

A. Implementasi Metode

Pada tahap ini dilakukan implementasi metode ke dalam bentuk aplikasi menggunakan bahasa pemrograman Java. Optimasi dilakukan untuk parameter α pada DES Brown dan α dan γ pada DES Holt yang nilainya berada di antara 0 dan 1. Optimasi parameter dilakukan untuk meminimumkan nilai MAPE. Berikut adalah langkah dalam penerapan DES Brown, DES Holt, *golden section*, dan *modified golden section*.

1. Metode pemulusan eksponensial ganda linear satu parameter dari Brown (DES Brown)

- Menset nilai parameter pemulusan α dengan ketentuan $0 < \alpha < 1$. Nilai α dipilih secara acak.
- Menginisialisasi nilai S' dan S'' pertama yakni $S'_1 = S''_1 = X_1$.
- Menghitung untuk satu periode berikutnya nilai
 - S' menggunakan persamaan (4);
 - S'' menggunakan persamaan (5);
 - a menggunakan persamaan (6); dan
 - b menggunakan persamaan (7).
- Melakukan prediksi satu periode berikutnya menggunakan persamaan (8).
- Mengulang langkah 3 dan 4 untuk seluruh kelompok data inisialisasi.
- Menggunakan nilai a dan b terakhir untuk prediksi beberapa periode ke depan.

- g. Menghitung nilai MAPE menggunakan persamaan (14) dengan membandingkan nilai prediksi dari langkah f dengan kelompok data pengujian.

$$MAPE = \frac{\sum_{i=1}^n |PE_i|}{n} \tag{14}$$

dengan

X_t = nilai data aktual periode t

F_t = nilai ramalan periode t

n = banyaknya data

$$PE_t = \left(\frac{X_t - F_t}{X_t} \right) 100$$

2. Metode pemulusan eksponensial ganda dua parameter dari Holt (DES Holt)

- a. Menset nilai parameter pemulusan α dan γ dengan ketentuan $0 < \alpha < 1$ dan $0 < \gamma < 1$. Nilai α dan γ dipilih secara acak.
- b. Menginisialisasi nilai S pertama yakni $S_1 = X_1$ dan b pertama menggunakan persamaan (12).
- c. Menghitung untuk satu periode berikutnya nilai S menggunakan persamaan (9) dan b menggunakan persamaan (10).
- d. Melakukan prediksi satu periode berikutnya menggunakan persamaan (11).
- e. Mengulang langkah 3 dan 4 untuk seluruh kelompok data inisialisasi.
- f. Menggunakan nilai S dan b terakhir untuk prediksi beberapa periode ke depan.
- g. Menghitung nilai MAPE menggunakan persamaan (14) dengan membandingkan nilai prediksi dari langkah f dengan kelompok data pengujian.

3. Metode golden section untuk optimasi parameter α pada DES Brown

- a. Menentukan selang ketidakpastian pertama $[a, d] = [0, 1]$ yang merupakan kendala dari optimasi yakni batas nilai α yaitu $0 < \alpha < 1$.
- b. Menghitung nilai b menggunakan persamaan (2) dan nilai c menggunakan persamaan (3).
- c. Menghitung nilai $f(b)$ dan $f(c)$ yang merupakan nilai MAPE data *testing* jika menggunakan nilai $\alpha = b$ dan $\alpha = c$.
- d. Menghitung $d - a$ dan membandingkannya dengan toleransi (ϵ).
Jika $d - a \leq \epsilon$ atau jumlah iterasi mencapai iterasi maksimum, iterasi selesai dan diperoleh nilai $b = c = \alpha$ (konvergen).
Jika tidak, lakukan langkah e kemudian ulangi langkah c dan d.
- e. Membandingkan nilai $f(b)$ dan $f(c)$ dan melakukan pembaruan selang dengan ketentuan sebagai berikut.

$$\text{Jika } f(b) < f(c), \quad \begin{cases} d = c, \\ c = b, \\ a = a, \\ b = ra + (1-r)d. \end{cases} \quad \text{Jika } f(b) \geq f(c), \quad \begin{cases} a = b, \\ b = c, \\ d = d, \\ c = a + d - b. \end{cases}$$

4. Modifikasi golden section untuk optimasi parameter α dan γ pada DES Holt

- a. Menentukan selang ketidakpastian pertama $[a_1, d_1] = [0, 1]$ yang merupakan kendala dari optimasi yakni batas nilai α yaitu $0 < \alpha < 1$ dan $[a_2, d_2] = [0, 1]$ yang merupakan kendala dari optimasi yakni $0 < \gamma < 1$.
- b. Menghitung nilai b_1 menggunakan persamaan (2) dan nilai c_1 menggunakan persamaan (3).
Menghitung nilai b_2 menggunakan persamaan (2) dan nilai c_2 menggunakan persamaan (3).
- c. Menghitung nilai $f(b_1, b_2), f(b_1, c_2), f(c_1, b_2)$, dan $f(c_1, c_2)$ yang merupakan nilai MAPE data *testing* jika menggunakan nilai $\alpha = b_1$ dan $\gamma = b_2, \alpha = b_1$ dan $\gamma = c_2$, dan seterusnya.

- d. Menghitung $d_1 - a_1$ dan $d_2 - a_2$ dan membandingkan dengan toleransi (ε).
Jika $d_1 - a_1 \leq \varepsilon$ atau $d_2 - a_2 \leq \varepsilon$ atau jumlah iterasi mencapai iterasi maksimum, iterasi selesai dan diperoleh nilai $b_1 = c_1 = \alpha$ dan $b_2 = c_2 = \gamma$ (konvergen).
Jika tidak, lakukan langkah 5 kemudian ulangi langkah 3 dan 4.
- e. Membandingkan nilai $f(b_1, b_2), f(b_1, c_2), f(c_1, b_2)$, dan $f(c_1, c_2)$ dan melakukan pembaruan selang dengan ketentuan berikut.

$$\begin{array}{l} \text{Jika } f(b_1, b_2) \text{ maksimum,} \\ \text{Jika } f(b_1, c_2) \text{ maksimum,} \\ \text{Jika } f(c_1, b_2) \text{ maksimum,} \\ \text{Jika } f(c_1, c_2) \text{ maksimum,} \end{array} \left\{ \begin{array}{l} a_1 = b_1, \\ b_1 = c_1, \\ d_1 = d_1, \\ c_1 = a_1 + d_1 - b_1, \\ a_2 = b_2, \\ b_2 = c_2, \\ d_2 = d_2, \\ c_2 = a_2 + d_2 - b_2. \\ \\ a_1 = b_1, \\ b_1 = c_1, \\ d_1 = d_1, \\ c_1 = a_1 + d_1 - b_1, \\ d_2 = c_2, \\ c_2 = b_2, \\ a_2 = a_2, \\ b_2 = ra_2 + (1-r)d_2. \\ \\ d_1 = c_1, \\ c_1 = b_1, \\ a_1 = a_1, \\ b_1 = ra_1 + (1-r)d_1, \\ a_2 = b_2, \\ b_2 = c_2, \\ d_2 = d_2, \\ c_2 = a_2 + d_2 - b_2. \\ \\ d_1 = c_1, \\ c_1 = b_1, \\ a_1 = a_1, \\ b_1 = ra_1 + (1-r)d_1, \\ d_2 = c_2, \\ c_2 = b_2, \\ a_2 = a_2, \\ b_2 = ra_2 + (1-r)d_2. \end{array} \right.$$

B. Perhitungan Kompleksitas

Perhitungan kompleksitas waktu dilakukan menggunakan pendekatan teoretis dan eksperimen. Pendekatan teoretis dilakukan dengan melakukan analisis jumlah operasi dalam algoritma, sedangkan pendekatan eksperimen dilakukan dengan melakukan uji coba untuk mengetahui *running time* algoritma menggunakan beberapa percobaan jumlah masukan n .

Pengujian *running time* dilakukan menggunakan laptop Dell Vostro 1088 dengan spesifikasi prosesor Intel Core 2 Duo 2.20 GHz dan RAM 4 GB DDR2 SDRAM 800 MHz. Perhitungan kompleksitas dilakukan terhadap implementasi metode DES Brown, DES Holt, *golden section*, dan *modified golden section*.

III. HASIL DAN PEMBAHASAN

A. Perhitungan Kompleksitas DES Brown

Tabel 2 menampilkan perhitungan jumlah operasi dalam implementasi DES Brown untuk mengetahui kompleksitas waktu teoretisnya.

Tabel 2: Perhitungan Jumlah Operasi Implementasi DES Brown

| Implementasi | Subjumlah | Jumlah Operasi |
|--|--------------|-----------------|
| St[0] = dataTraining[0]; | 1 | 11 |
| Stt[0] = dataTraining[0]; | 1 | |
| at[0] = 0; | 1 | |
| bt[0] = 0; | 1 | |
| Ft[0] = 0; | 1 | |
| Ft[1] = 0; | 1 | |
| Et[0] = 0; | 1 | |
| PEt[0] = 0; | 1 | |
| APEt[0] = 0; | 1 | |
| jmlAPEt = 0; | 1 | |
| MAPE = 0; | 1 | |
| for (i = 1; i < dataTraining.length; i++) { | $(4/5)n - 1$ | $5((4/5)n - 1)$ |
| St[i] = (alfa * dataTraining[i] + ((1 - alfa) * St[i - 1])); | 1 | $= 4n - 5$ |
| Stt[i] = (alfa * Stt[i] + ((1 - alfa) * Stt[i - 1])); | 1 | |
| at[i] = 2 * St[i] - Stt[i]; | 1 | |
| bt[i] = (alfa / (1 - alfa)) * (St[i] - Stt[i]); | 1 | |
| Ft[i + 1] = at[i] + bt[i] * 1; | 1 | |
| } | | |
| for (j = 1; j <= dataTesting.length; j++, i++) { | $(1/5)n$ | $5(1/5)n$ |
| Ft[i] = at[dataTraining.length - 1] + bt[dataTraining.length - 1] * j; | 1 | $= n$ |
| Et[i] = dataTesting[j - 1] - Ft[i]; | 1 | |
| PEt[i] = (Et[i] / dataTesting[j - 1]) * 100; | 1 | |
| APEt[i] = abs(PEt[i]); | 1 | |
| jmlAPEt = jmlAPEt + APEt[i]; | 1 | |
| } | | |
| MAPE = jmlAPEt / dataTesting.length; | 1 | 1 |

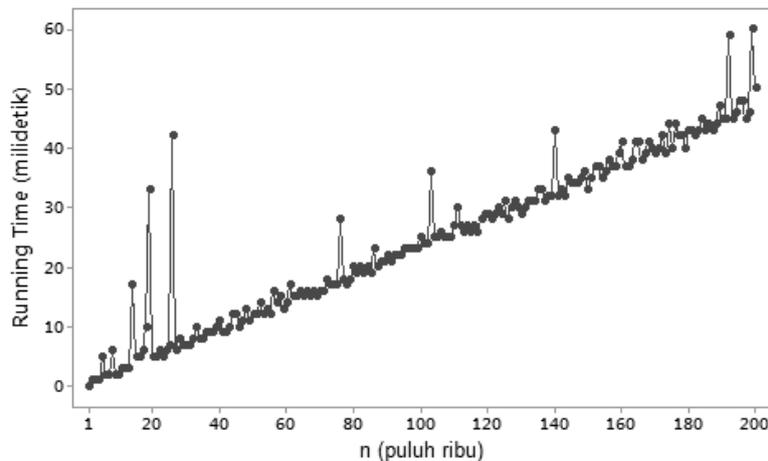
Berdasarkan perhitungan jumlah operasi diperoleh kompleksitas waktu berikut.

$$T(n) = 11 + 4n - 5 + n + 1 = 5n + 7$$

Kompleksitas tersebut sama untuk kasus *worst case* dan *best case*, atau dapat ditulis dalam notasi asimtotik $T(n) = O(n) = \Omega(n)$.

Selanjutnya dilakukan perhitungan *running time* untuk beberapa jumlah data n . Gambar 2 menunjukkan grafik pertambahan *running time* eksekusi DES Brown untuk $n = 10000$ (sepuluh ribu) s.d. $n = 2000000$ (dua juta) dengan laju pertambahan $n = 10000$ (sepuluh ribu) per percobaan. Grafik menunjukkan bahwa pertambahan *running time* sebanding dengan pertambahan n . Hal ini sesuai dengan hasil analisis teoretis yang menunjukkan

bahwa DES Brown memiliki kompleksitas waktu $O(n)$.



Gambar 2: Grafik Pertambahan *Running Time* DES Brown untuk Beberapa Jumlah Data n

B. Perhitungan Kompleksitas DES Holt

Tabel 3 menampilkan perhitungan jumlah operasi dalam implementasi DES Holt untuk mengetahui kompleksitas waktu teoretisnya.

Tabel 3: Perhitungan Jumlah Operasi Implementasi DES Holt

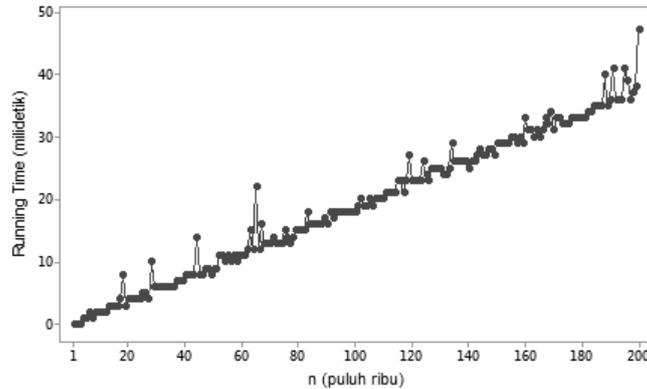
| Implementasi | Subjumlah | Jumlah Operasi |
|--|--------------|-----------------|
| $St[0] = dataTraining[0];$ | 1 | 9 |
| $bt[0] = dataTraining[1] - dataTraining[0];$ | 1 | |
| $Ft[0] = 0;$ | 1 | |
| $Ft[1] = 0;$ | 1 | |
| $Et[0] = 0;$ | 1 | |
| $PEt[0] = 0;$ | 1 | |
| $APEt[0] = 0;$ | 1 | |
| $jmlAPEt = 0;$ | 1 | |
| $MAPE = 0;$ | 1 | |
| for ($i = 1; i < dataTraining.length; i++$) { | $(4/5)n - 1$ | $3((4/5)n - 1)$ |
| $St[i] = alfa * dataTraining[i] + (1 - alfa) * (St[i - 1] + bt[i - 1]);$ | 1 | $= (12/5)n - 3$ |
| $bt[i] = gama * (St[i] - St[i - 1]) + (1 - gama) * bt[i - 1];$ | 1 | |
| $Ft[i + 1] = St[i] + bt[i] * 1;$ | 1 | |
| } | | |
| for ($j = 1; j <= dataTesting.length; j++, i++$) { | $(1/5)n$ | $5(1/5)n$ |
| $Ft[i]=St[dataTraining.length - 1] + bt[dataTraining.length - 1] * j;$ | 1 | $= n$ |
| $Et[j] = dataTesting[j - 1] - Ft[j];$ | 1 | |
| $PEt[j] = (Et[j] / dataTesting[j - 1]) * 100;$ | 1 | |
| $APEt[j] = abs(PEt[j]);$ | 1 | |
| $jmlAPEt = jmlAPEt + APEt[j];$ | 1 | |
| } | | |
| $MAPE = jmlAPEt / dataTesting.length;$ | 1 | 1 |

Berdasarkan perhitungan jumlah operasi diperoleh kompleksitas waktu berikut.

$$T(n) = 9 + \frac{12}{5}n - 3 + n + 1 = \frac{17}{5}n + 7$$

Kompleksitas tersebut sama untuk kasus *worst case* dan *best case*, atau dapat ditulis dalam notasi asimtotik $T(n) = O(n) = \Omega(n)$.

Lalu dilakukan perhitungan *running time* untuk beberapa jumlah data n . Gambar 3 menunjukkan grafik pertambahan *running time* eksekusi DES Holt untuk $n = 10000$ (sepuluh ribu) s.d. $n = 2000000$ (dua juta) dengan laju pertambahan $n = 10000$ (sepuluh ribu) per percobaan. Grafik menunjukkan bahwa pertambahan *running time* sebanding dengan pertambahan n . Hal ini sesuai dengan hasil analisis teoretis yang menunjukkan bahwa DES Holt memiliki kompleksitas waktu $O(n)$.



Gambar 3: Grafik Pertambahan *Running Time* DES Holt untuk Beberapa Jumlah Data n

C. Perhitungan Kompleksitas Metode *Golden Section*

Tabel 4 menampilkan perhitungan jumlah operasi dalam implementasi metode *golden section* untuk mengetahui kompleksitas waktu teoretisnya.

Tabel 4: Perhitungan Jumlah Operasi Implementasi Metode *Golden Section*

| Implementasi | Subjumlah | Jumlah Operasi |
|---|--|--|
| $a = 0;$ | 1 | 8 |
| $d = 1;$ | 1 | |
| $\epsilon = 0.00000001;$ | 1 | |
| $iter = 50;$ | 1 | |
| $r = (-1 + \sqrt{5}) / 2;$ | 1 | |
| $k = 0;$ | 1 | |
| $b = r * a + (1 - r) * d;$ | 1 | |
| $c = a + d - b;$ | 1 | |
| $while ((abs(d - a) > \epsilon) \&\& (k < iter)) \{$ | Pada <i>while</i> , 3 atau 1. <i>Looping</i> $\lceil \log_r \epsilon \rceil$ kali | 3 atau $(10n + 20) * \log \epsilon$ |
| $k = k + 1;$ | 1 | |
| $fb = \text{hitung}(b); //DES\ Brown\ dengan\ alfa = b$ | $5n + 7$ | |
| $fc = \text{hitung}(c); //DES\ Brown\ dengan\ alfa = c$ | $5n + 7$ | |
| $if (fb < fc) \{$ | 1 | 4 |
| $d = c;$ | 1 | |
| $c = b;$ | 1 | |
| $b = r * a + (1 - r) * d;$ | 1 | |
| $\} else \{$ | | |
| $a = b;$ | 1 | |
| $b = c;$ | 1 | |
| $c = a + d - b;$ | 1 | |
| $\}$ | | |
| $\}$ | | |

Besarnya selisih $|d - a|$ pada iterasi k atau L_k dapat diketahui menggunakan rumus berikut [12].

$$L_k = r^k |d - a| = r^k L_0 \tag{15}$$

Dengan menyelesaikan persamaan (15), jumlah iterasi pada *while* dapat diketahui menggunakan formula (16).

$$k = \lceil \log_r \frac{\epsilon}{L_0} \rceil \tag{16}$$

Karena $L_0 = 1$, maka $k = \lceil \log \varepsilon \rceil$.

Berdasarkan perhitungan jumlah operasi diperoleh kompleksitas waktu untuk *best case* adalah

$$T(n) = 8 + 3 = 11$$

atau dalam notasi asimtotik

$$T(n) = \Omega(1)$$

sedangkan untuk *worst case* adalah

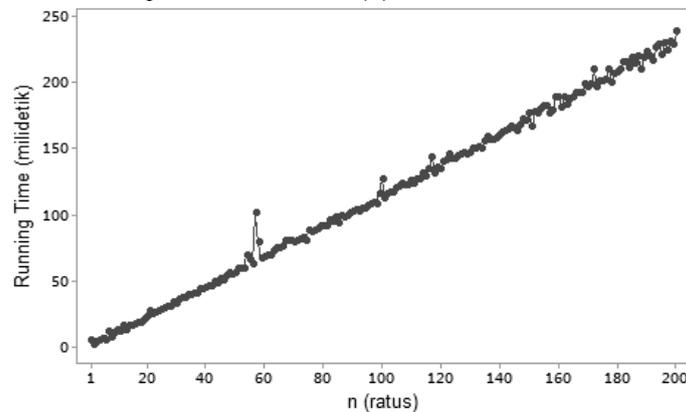
$$T(n) = 8 + (10n + 20)(\lceil \log \varepsilon \rceil) = 8 + 10n(\lceil \log \varepsilon \rceil) + 20(\lceil \log \varepsilon \rceil)$$

atau dalam notasi asimtotik ditulis

$$T(n) = O(n)$$

karena $\lceil \log \varepsilon \rceil$ konstan.

Selanjutnya dilakukan perhitungan *running time* untuk beberapa jumlah data n . Gambar 4 menunjukkan grafik pertambahan *running time* eksekusi metode *golden section* untuk $n = 100$ s.d. $n = 20000$ (dua puluh ribu) dengan laju pertambahan $n = 100$ per percobaan. Grafik menunjukkan bahwa pertambahan *running time* sebanding dengan pertambahan n . Hal ini sesuai dengan hasil analisis teoretis yang menunjukkan bahwa metode *golden section* memiliki kompleksitas waktu $O(n)$.



Gambar 4: Grafik Pertambahan *Running Time* Metode *Golden Section* untuk Beberapa Jumlah Data n

D. Perhitungan Kompleksitas Metode *Modified Golden Section*

Tabel 5 menampilkan perhitungan jumlah operasi dalam implementasi metode *golden section* untuk mengetahui kompleksitas waktu teoretisnya. Berdasarkan perhitungan jumlah operasi diperoleh kompleksitas waktu untuk *best case* adalah

$$T(n) = 12 + 5 = 17$$

atau dalam notasi asimtotik

$$T(n) = \Omega(1)$$

sedangkan untuk *worst case* adalah

$$T(n) = 12 + \left(\frac{68}{5}n + 38\right)(\lceil \log \varepsilon \rceil) = 12 + \frac{68}{5}n(\lceil \log \varepsilon \rceil) + 38(\lceil \log \varepsilon \rceil)$$

atau dalam notasi asimtotik ditulis

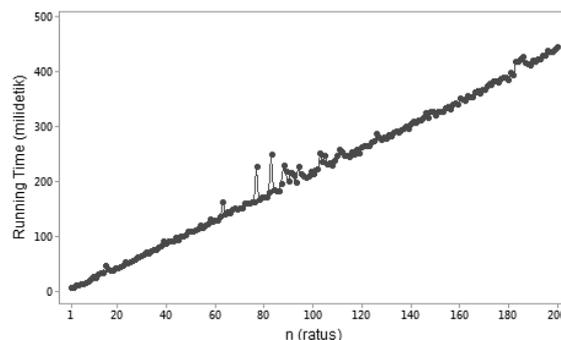
$$T(n) = O(n)$$

karena $\lceil \log \varepsilon \rceil$ konstan.

Tabel 5: Perhitungan Jumlah Operasi Implementasi *Modified Golden Section*

| Implementasi | Subjumlah | Jumlah Operasi |
|--|---|---|
| a1 = 0; | 1 | 12 |
| a2 = 0; | 1 | |
| d1 = 1; | 1 | |
| d2 = 1; | 1 | |
| epsilon = 0.00000001; | 1 | |
| iter = 50; | 1 | |
| r = (-1 + sqrt(5)) / 2; | 1 | |
| k = 0; | 1 | |
| b1 = r * a1 + (1 - r) * d1; | 1 | |
| b2 = r * a2 + (1 - r) * d2; | 1 | |
| c1 = a1 + d1 - b1; | 1 | |
| c2 = a2 + d2 - b2; | 1 | |
| while ((abs(d1 - a1) > epsilon) && (abs(d2 - a2) > epsilon) && (k < iter)) { | Pada while, 5 atau 1. Looping-nya [log _r ε] kali | 5 atau ((68/5)n + 36) * log ε atau ((68/5)n + 37) * log ε atau ((68/5)n + 38) * log ε |
| k = k + 1; | 1 | |
| fb1b2 = hitung(b1, b2); //DES Brown dengan alfa = b1, gama = b2 | (17/5)n + 7 | |
| fb1c2 = hitung(b1, c2); //DES Brown dengan alfa = b1, gama = c2 | (17/5)n + 7 | |
| fc1b2 = hitung(c1, b2); //DES Brown dengan alfa = c1, gama = b2 | (17/5)n + 7 | |
| fc1c2 = hitung(c1, c2); //DES Brown dengan alfa = c1, gama = c2 | (17/5)n + 7 | |
| if (fb1b2 > fb1c2 && fb1b2 > fc1b2 && fb1b2 > fc1c2) { | 1 | 7 8 9 9 |
| a1 = b1; | 1 | |
| b1 = c1; | 1 | |
| c1 = a1 + d1 - b1; | 1 | |
| a2 = b2; | 1 | |
| b2 = c2; | 1 | |
| c2 = a2 + d2 - b2; | 1 | |
| } else { | | |
| if (fb1c2 > fb1b2 && fb1c2 > fc1b2 && fb1c2 > fc1c2) { | 1 | |
| a1 = b1; | 1 | |
| b1 = c1; | 1 | |
| c1 = a1 + d1 - b1; | 1 | |
| d2 = c2; | 1 | |
| c2 = b2; | 1 | |
| b2 = r * a2 + (1 - r) * d2; | 1 | |
| } else { | | |
| if (fc1b2 > fb1b2 && fc1b2 > fb1c2 && fc1b2 > fc1c2) { | 1 | |
| d1 = c1; | 1 | |
| c1 = b1; | 1 | |
| b1 = r * a1 + (1 - r) * d1; | 1 | |
| a2 = b2; | 1 | |
| b2 = c2; | 1 | |
| c2 = a2 + d2 - b2; | 1 | |
| } else { | | |
| d1 = c1; | 1 | |
| c1 = b1; | 1 | |
| b1 = r * a1 + (1 - r) * d1; | 1 | |
| d2 = c2; | 1 | |
| c2 = b2; | 1 | |
| b2 = r * a2 + (1 - r) * d2; | 1 | |
| } | | |
| } | | |
| } | | |
| } | | |

Selanjutnya dilakukan perhitungan *running time* untuk beberapa jumlah data n . Gambar 5 menunjukkan grafik pertambahan *running time* eksekusi metode *golden section* untuk $n = 100$ s.d. $n = 20000$ (dua puluh ribu) dengan laju pertambahan $n = 100$ per percobaan. Grafik menunjukkan bahwa pertambahan *running time* sebanding dengan pertambahan n . Hal ini sesuai dengan hasil analisis teoretis yang menunjukkan bahwa metode *modified golden section* memiliki kompleksitas waktu $O(n)$.



Gambar 5: Grafik Pertambahan *Running Time* Metode *Modified Golden Section* untuk Beberapa Jumlah Data n

IV. KESIMPULAN

Perhitungan kompleksitas waktu menggunakan pendekatan teoretis dan eksperimen telah dilakukan terhadap metode DES Brown, DES Holt, *golden section* dan *modified golden section* dan kesemua metode memiliki kompleksitas waktu linear atau $O(n)$, dengan n adalah banyaknya data. Algoritma yang memiliki waktu linear biasanya terdapat pada kasus yang setiap elemen masukannya dikenai proses yang sama. Bila n dijadikan dua kali semula, waktu pelaksanaan algoritma juga dua kali semula.

Linearitas waktu terjadi karena proses implementasi tidak memiliki *loop* bersarang. Meskipun pada implementasi metode *golden section* dan *modified golden section* memiliki jumlah *looping* yang logaritmik, namun nilainya adalah konstan sehingga tidak berpengaruh pada kompleksitas waktunya.

Secara umum metode DES Brown dan DES Holt memiliki *running time* lebih kecil dibandingkan metode *golden section* dan *modified golden section*. Hal ini terjadi karena dalam pencarian parameter optimal, kedua metode *golden section* beberapa kali memanggil metode DES Brown dan DES Holt. Proses percobaan berulang-ulang inilah yang menyebabkan adanya perbedaan waktu signifikan.

Penelitian selanjutnya selain menghitung kompleksitas waktu diharapkan juga dapat melakukan perhitungan kompleksitas ruang memori.

DAFTAR PUSTAKA

- [1] N. D. Saputra, A. Aziz, and B. Harjito, "Parameter Optimization of Brown's and Holt's Double Exponential Smoothing Using Golden Section Method for Predicting Indonesian Crude Oil Price (ICP)," in *Proceedings of 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2016, p. 352—356.
- [2] R. Munir, *Metode Numerik, Revisi Ketiga*. Bandung: Penerbit Informatika, 2013.
- [3] D. W. Nugraha, "Penerapan Kompleksitas Waktu Algoritma Prim untuk Menghitung Kemampuan Komputer dalam Melaksanakan Perintah," *J. Ilm. Foristek*, vol. 2, no. 2, p. 195—207, 2012.
- [4] D. al Mahkya, H. Yasin, and M. A. Mukid, "Aplikasi Metode Golden Section untuk Optimasi Parameter pada Metode Exponential Smoothing," *J. Gaussian*, vol. 3, no. 4, p. 605—614, 2014.
- [5] N. Yuwida, L. Hanafi, and N. Wahyuningsih, "Estimasi Parameter Alfa dan Gama dalam Pemulusan Eksponensial Ganda Dua Parameter dengan Metode Modifikasi Golden Section," *J. Sains dan Seni ITS*, vol. 1, no. 1, p. A-18—A-22, 2012.
- [6] M. F. Qudratullah, "Prediksi Harga Rata-Rata Minyak Mentah Indonesia (Indonesian Crude Oil Price/ICP) Menggunakan Berbagai Teknik Analisis Data Timeseries dan Implikasinya Terhadap Harga Bahan Bakar Minyak (BBM) Bersubsidi di Indonesia," *Integr. Lab J.*, vol. I, no. 1, p. 1—19, 2013.
- [7] R. Halimi, W. Anggraeni, and R. Tyasnurita, "Pembuatan Aplikasi Peramalan Jumlah Permintaan Produk dengan Metode Time Series Exponential Smoothing Holts Winter di PT Telekomunikasi Indonesia Tbk," *J. Tek. POMITS*, vol. 1, no. 1, p. 1—6, 2013.
- [8] A. Yulitasari, "Perbandingan Metode Pemulusan Eksponensial Ganda Holt dengan Metode Pemulusan Eksponensial Ganda Brown," Universitas Negeri Semarang, 2011.
- [9] P. S. Kalekar, "Time series Forecasting using Holt-Winters Exponential Smoothing," Mumbai, 2004.
- [10] S. Makridakis, S. C. Wheelwright, V. E. McGee, U. S. Andriyanto, and A. Basith, *Metode dan Aplikasi Peramalan Jilid 1, Edisi Kedua*. Jakarta: Penerbit Erlangga, 1995.
- [11] D. Luknanto, "Pengantar Optimasi Nonlinear, Bahan Penataran Jurusan Teknik Sipil," Yogyakarta, 2000.
- [12] E. Sumarminingsih, "Algoritma Golden Section Search untuk Mencari Solusi Optimal pada Pemrograman Non Linear Tanpa Kendala, Bahan Kuliah Jurusan Matematika," Malang, 2011.