

**LAPORAN TUGAS BESAR
ANALISIS KOMPLEKSITAS ALGORITMA**

**“Perbandingan Efisiensi Algoritma Bubble Sort dan
Insertion Sort pada Pengurutan Rating Gofood”**



Oleh:

**Ghaisani Zhafarina(103012300379)
Meyra Firdhaus Isnaini(103012300179)**

**Program Studi S-1 Informatika
Fakultas Informatika
Telkom University
20224**

A. Deskripsi Topik

Studi ini bertujuan untuk membandingkan dua algoritma pengurutan sederhana, yaitu Bubble Sort dan Insertion Sort, dalam mengurutkan data rating makanan dari restoran pada platform Gofood. Analisis akan menunjukkan bagaimana teori kompleksitas waktu $O(n^2)$ diterapkan dalam praktik dan algoritma mana yang lebih cepat dalam situasi tertentu, termasuk pada data yang hampir terurut. Dengan membandingkan kedua algoritma, studi ini memberikan wawasan tentang kelebihan dan kekurangan masing-masing serta situasi mana yang lebih cocok untuk digunakan dalam pengurutan rating makanan.

Kami akan melakukan perbandingan antara dua algoritma pengurutan yaitu Bubble Sort sebagai algoritma iteratif dan Insertion Sort sebagai algoritma rekursif dalam konteks pengurutan data rating makanan dari restoran yang ada di Yogyakarta pada platform Gofood.

Dalam studi kasus ini, implementasi Bubble Sort menggunakan algoritma iteratif, yang mana setiap elemen dibandingkan dengan elemen berikutnya secara berulang pada setiap iterasi untuk memastikan data terurut. Sementara itu, implementasi Insertion Sort menggunakan algoritma rekursif, di mana algoritman ini bekerja dengan memecah masalah pengurutan menjadi sub-masalah lebih kecil. Setiap elemen dibandingkan dengan elemen-elemen sebelumnya dan disisipkan pada posisi yang tepat menggunakan pemanggilan fungsi rekursif.

B. Deskripsi Algoritma Bubble Sort

Bubble Sort adalah algoritma pengurutan yang bekerja dengan cara menukar elemen-elemen yang berdekatan secara berulang-ulang jika berada dalam urutan yang salah. Dengan menyapu kumpulan data beberapa kali, nilai yang lebih kecil “bubble up” atau “menggelembung” secara bertahap ke arah awal, sementara nilai yang lebih besar bergeser ke arah akhir. Bubble Sort ini hanya dapat diimplementasi dengan cara iteratif.

C. Deskripsi Algoritma Insertion Sort

Insertion Sort adalah algoritma pengurutan yang bekerja dengan menyisipkan secara berulang-ulang setiap elemen dari daftar yang belum diurutkan ke dalam posisi yang benar di bagian daftar yang telah diurutkan. Cara kerja penyisipan pada algoritma ini mirip dengan penyortiran kartu remi di tangan. Diasumsikan bahwa kartu pertama sudah diurutkan dalam permainan kartu, lalu kita memilih kartu yang belum diurutkan. Jika kartu yang tidak diurutkan yang dipilih lebih besar dari kartu pertama, maka kartu tersebut akan ditempatkan di sisi kanan. Jika tidak, maka kartu

tersebut akan ditempatkan di sisi kiri. Demikian pula, semua kartu yang belum disortir diambil dan diletakkan di tempat yang tepat. Insertion dapat diimplementasikan secara iteratif dan rekursif. Gambar grafik perbandingan bisa berbeda karena adanya keterbatasan pada CPU laptop dan lain hal.

D. Algoritma

<https://github.com/gzhfrn/TUGAS-BESAR-AKA>

Bubble Sort Iteratif:

```
// BubbleSort untuk mengurutkan berdasarkan rating
func BubbleSort(data [][]string) [][]string {
    n := len(data)
    for i := 0; i < n-1; i++ {
        for j := 0; j < n-i-1; j++ {
            rating1, _ := strconv.ParseFloat(data[j][0], 64)
            rating2, _ := strconv.ParseFloat(data[j+1][0], 64)
            if rating1 < rating2 { // Tukar jika rating1 lebih kecil
                data[j], data[j+1] = data[j+1], data[j]
            }
        }
    }
    return data
}
```

Kompleksitas waktu:

$$T(n) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-i-1} 1$$

$$T(n) = \sum_{i=0}^{n-1} (n - i - 1)$$

$$T(n) = \sum_{i=0}^{n-1} (n - i) = \sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} i$$

$$T(n) = n \cdot n - \frac{(n-1) \cdot n}{2}$$

$$T(n) = n^2 - \frac{n^2 - n}{2}$$

$$T(n) = \frac{2n^2 - (n^2 - n)}{2}$$

$$T(n) = \frac{2n^2 - n^2 + n}{2}$$

$$T(n) = \frac{n^2 + n}{2}$$

$$T(n) = O(n^2)$$

Insertion Sort Rekursif:

```

// Rekursif insertion sort untuk menyisipkan elemen ke posisi yang tepat
func insert(data [][]string, i int) {    insert redeclared in this block
    if i <= 0 {
        return
    }

    key := data[i]
    j := i - 1

    // Geser elemen ke kanan jika lebih kecil dari elemen sebelumnya
    for j >= 0 {
        rating1, _ := strconv.ParseFloat(data[j][0], 64)
        ratingKey, _ := strconv.ParseFloat(key[0], 64)
        if rating1 < ratingKey {
            data[j+1] = data[j]
            j--
        } else {
            break
        }
    }
    data[j+1] = key
}

// Rekursif insertion sort untuk mengurutkan data
func RecursiveInsertionSort(data [][]string, i int) [][]string {    Recursi
    if i == len(data) { // Basis kasus
        return data
    }
    // Sisipkan elemen ke tempat yang sesuai
    insert(data, i)    cannot use data (variable of type [][]st
    // Rekursi ke elemen berikutnya
    return RecursiveInsertionSort(data, i+1)    RecursiveInsertionSort(da
}

```

Kompleksitas Waktu:

Insertion Sort Rekursif:

A. Relasi Rekurensi:

$$T(n) = T(n-1) + c \cdot n$$

B. Menentukan Solusi Homogen:

$$T(n) - T(n-1) = 0$$

Persamaan karakteristik:

$$r - 1 = 0$$

$$r=1$$

Maka, solusi homogen:

$$T_h(n)=C_1$$

C. Menentukan Solusi non homogen:

Fungsi non homogen:

$$T_p(n) = A \cdot n + B$$

D. Substitusi ke relasi rekursi:

$$T(n) = T(n-1) + c \cdot n$$

Substitusi $T_p(n)$ ke persamaan:

$$A \cdot n + B = A \cdot (n-1) + B + c \cdot n$$

$$A + c = A \Rightarrow A = c$$

$$B - A = B \Rightarrow A = 0$$

Jadi, solusi non homogennya adalah

$$T_p(n) = c \cdot n$$

E. Menggabungkan Solusi homogen dan non homogen:

$$T(n) = T_h(n) + T_p(n)$$

$$T(n) = C_1 + c \cdot n$$

$$T(n) = C_1 + c \cdot n$$

F. Menggunakan Kondisi Awal untuk Menentukan Konstanta

Kondisi awal:

$$T(1) = O(1)$$

Substitusi

$$T(1) = C_1 + c:$$

$$1 = C_1 + c$$

Maka:

$$C_1 = 1 - c$$

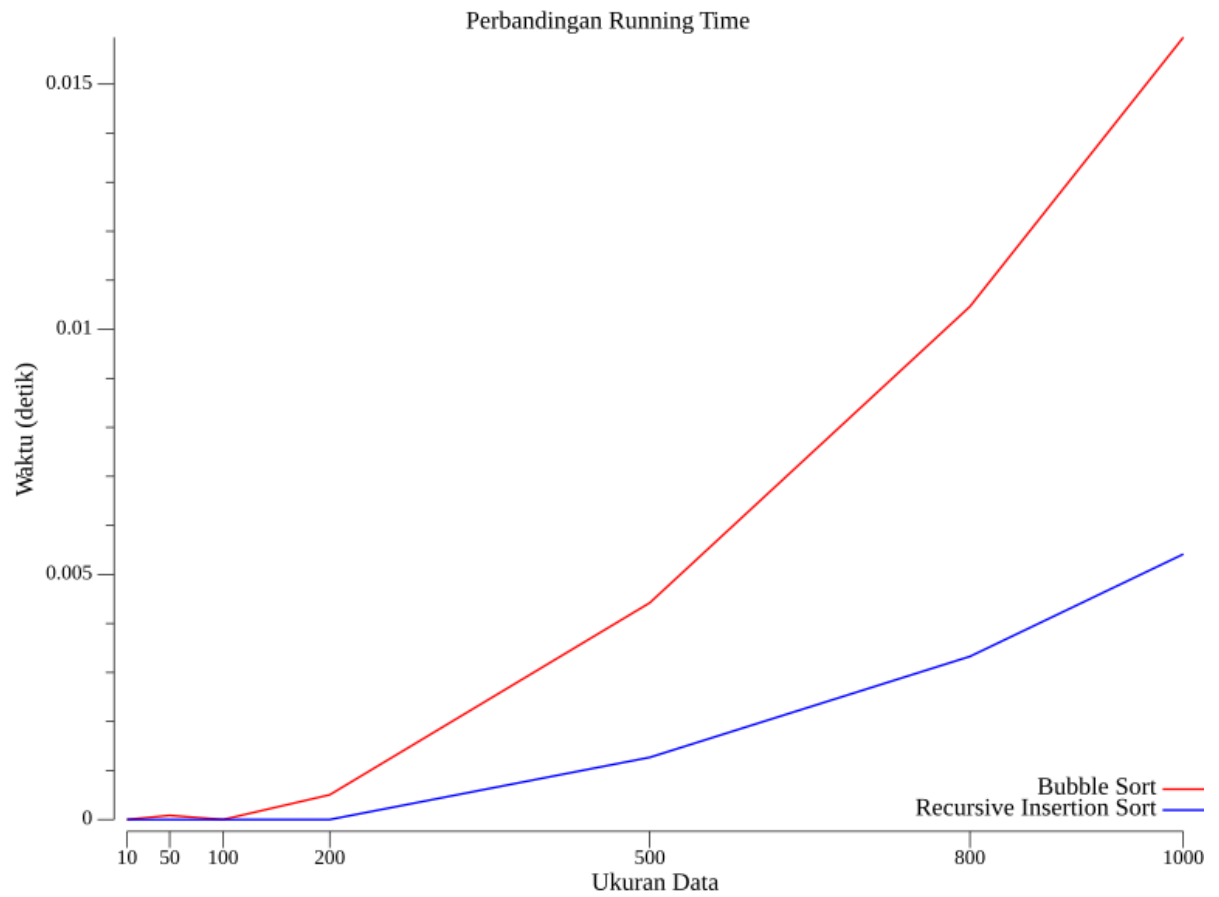
Sehingga solusi akhirnya:

$$T(n) = (1 - c) + c \cdot n$$

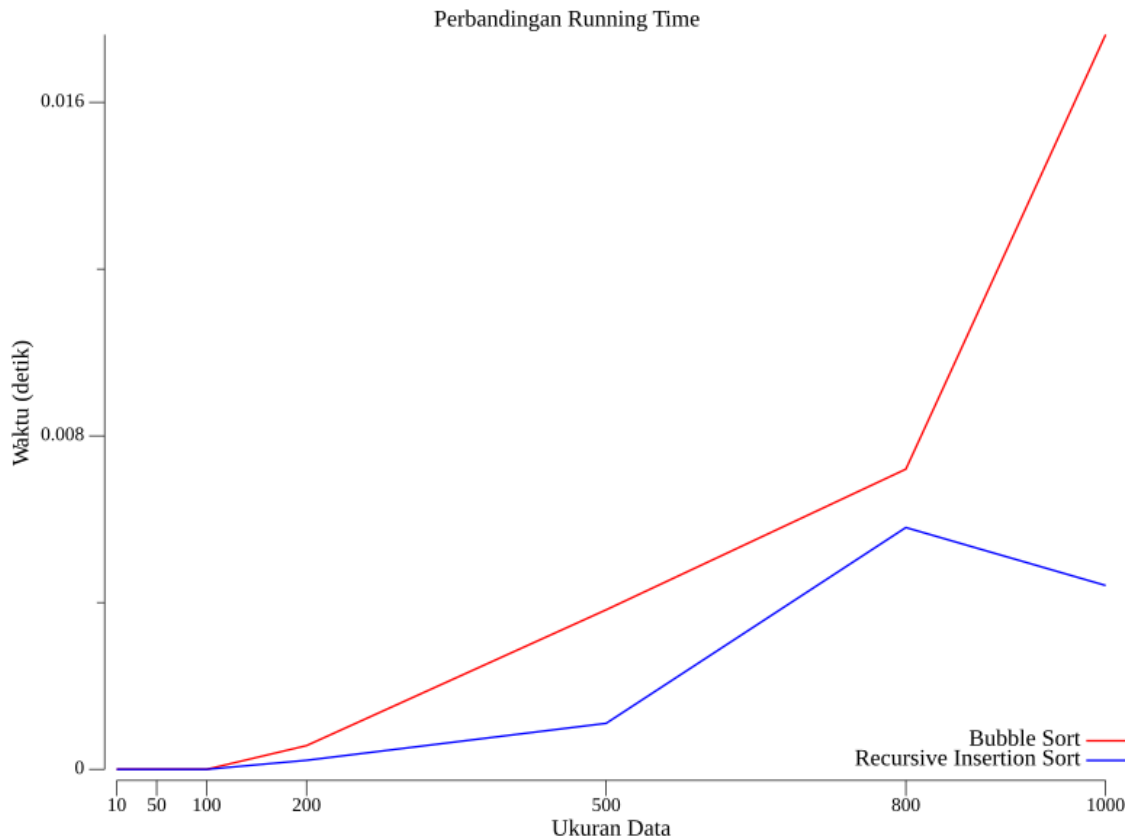
#Kompleksitas Waktu Asimtotik:

$$T(n) = O(n^2)$$

E. Grafik Perbandingan *running time*



Grafik perbandingan running time 1



Grafik perbandingan running time 2

Gambar grafik perbandingan dapat berbeda karena adanya keterbatasan pada sistem operasi atau lingkungan runtime yang mempengaruhi waktu eksekusi. Salah satu penyebabnya adalah kapasitas CPU atau memori yang terbatas. Jika ada aplikasi lain yang berjalan secara bersamaan, sumber daya yang tersedia dapat berkurang.

F. Analisis Perbandingan


```
C:\Users\Ghaisani\Documents\Telkom\Semester 3\AKA\coba2>go run rn.go
Membandingkan running time Bubble Sort dan Recursive Insertion Sort...
Ukuran data: 10
Bubble Sort: 0.0000 detik | Insertion Sort: 0.0000 detik

Ukuran data: 50
Bubble Sort: 0.0000 detik | Insertion Sort: 0.0000 detik

Ukuran data: 100
Bubble Sort: 0.0000 detik | Insertion Sort: 0.0000 detik

Ukuran data: 200
Bubble Sort: 0.0001 detik | Insertion Sort: 0.0006 detik

Ukuran data: 500
Bubble Sort: 0.0042 detik | Insertion Sort: 0.0005 detik

Ukuran data: 800
Bubble Sort: 0.0113 detik | Insertion Sort: 0.0036 detik

Ukuran data: 1000
Bubble Sort: 0.0149 detik | Insertion Sort: 0.0037 detik

Grafik berhasil disimpan sebagai grafikrn7.png
```

Berdasarkan hasil running time perbandingan dua algoritma bubble sort iteratif dan insertion sort rekursif untuk ukuran data kecil (10, 50, 100) waktu eksekusi untuk kedua algoritma sama-sama sangat kecil (0.0000 detik), ini menunjukkan bahwa untuk dataset kecil, perbedaan performa antara Bubble Sort dan Insertion Sort tidak signifikan karena beban komputasi rendah.

Ukuran data sedang (200, 500), pada ukuran data 200, Bubble Sort menunjukkan waktu eksekusi yang sedikit lebih cepat dibandingkan Insertion Sort (0.0001 detik vs. 0.0006 detik). Namun, pada ukuran data 500, Insertion Sort lebih cepat daripada Bubble Sort

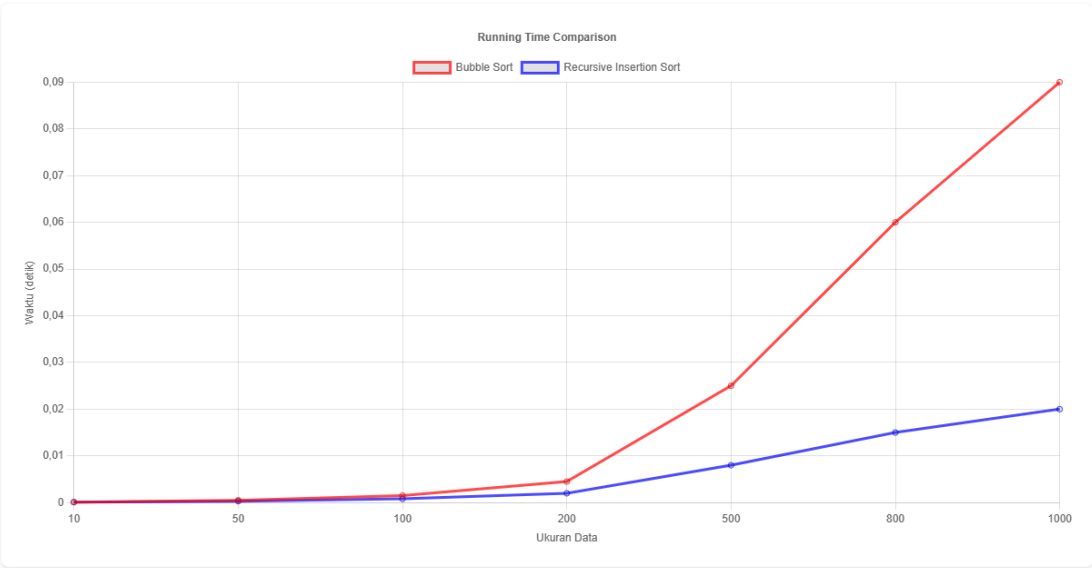
Ukuran data besar (800, 1000), pada ukuran data 800 dan 1000, Insertion Sort lebih cepat dibandingkan Bubble Sort. Misalnya, pada ukuran 1000, Bubble Sort membutuhkan 0.0149 detik sementara Insertion Sort membutuhkan 0.0037 detik (lebih cepat).

Dari analisis diatas dapat disimpulkan bahwa Bubble Sort lebih cocok untuk ukuran data yang lebih kecil dan Insertion Sort lebih cocok untuk ukuran data yang lebih besar.

G. Implementasi

Hasil Pengurutan Top 20 Restaurant

Ukuran Data	Bubble Sort (detik)	Recursive Insertion Sort (detik)
10	0.0001	0.0001
50	0.0005	0.0003
100	0.0015	0.0008
200	0.0045	0.0020
500	0.0250	0.0080
800	0.0600	0.0150
1000	0.0900	0.0200



Bubble Sort Rankings

1. SRI GUSTINI (4.9)
2. Trihanggo; Kopi Trip (4.9)
3. Kaliurang; Fore Coffee (4.9)
4. Muryanti (4.9)

Recursive Insertion Sort Rankings

1. SRI GUSTINI (4.9)
2. Trihanggo; Kopi Trip (4.9)
3. Kaliurang; Fore Coffee (4.9)
4. Muryanti (4.9)

Bubble Sort Rankings

1. SRI GUSTINI (4.9)
2. Trihanggo; Kopi Trip (4.9)
3. Kaliurang; Fore Coffee (4.9)
4. Muryanti (4.9)
5. Demangan Baru; Eatlah General (4.9)
6. Malangrejo; Ada Food & Juice (4.9)
7. Katamso; Terang Mulyo New Kitchen & Juice Bar (4.9)
8. Gang Sartono; Kedai 57 Yk (4.9)
9. Yk; DF HEALTHY KITCHEN (4.9)
10. Sadiman (4.9)
11. Gamping; Gubug Makan Mallindo Online (4.9)
12. Karang Mloko; Pengilon (4.8)
13. Gamping;Penyetan Mmd (4.8)
14. Jogja City Mall; Mixue Maranatha Surya Sumantri (4.8)
15. Ring Road Utara; Bebek Goreng H.Slamet Jogja (4.8)
16. Andreas Nanang Prasetya (4.8)
17. Sinduadi; Waroeng Geprek Rahayu (4.8)
18. Palagan Tentara Pelajar; Nasi Pecel Yu Sri (4.8)
19. Palagan Tentara Pelajar; Rumah Makan & Seafood 99 Migrate (4.8)
20. Palagan;Es Teh Indonesia General (4.8)

Recursive Insertion Sort Rankings

1. SRI GUSTINI (4.9)
2. Trihanggo; Kopi Trip (4.9)
3. Kaliurang; Fore Coffee (4.9)
4. Muryanti (4.9)
5. Demangan Baru; Eatlah General (4.9)
6. Malangrejo; Ada Food & Juice (4.9)
7. Katamso; Terang Mulyo New Kitchen & Juice Bar (4.9)
8. Gang Sartono; Kedai 57 Yk (4.9)
9. Yk; DF HEALTHY KITCHEN (4.9)
10. Sadiman (4.9)
11. Gamping; Gubug Makan Mallindo Online (4.9)
12. Karang Mloko; Pengilon (4.8)
13. Gamping;Penyetan Mmd (4.8)
14. Jogja City Mall; Mixue Maranatha Surya Sumantri (4.8)
15. Ring Road Utara; Bebek Goreng H.Slamet Jogja (4.8)
16. Andreas Nanang Prasetya (4.8)
17. Sinduadi; Waroeng Geprek Rahayu (4.8)
18. Palagan Tentara Pelajar; Nasi Pecel Yu Sri (4.8)
19. Palagan Tentara Pelajar; Rumah Makan & Seafood 99 Migrate (4.8)
20. Palagan;Es Teh Indonesia General (4.8)

H. Referensi

<https://www.kaggle.com/datasets/iannarsa/gofood-merchant-on-yogyakarta>
<https://learnersbucket.com/examples/algorithms/recursive-insertion-sort-algorithm/>
https://en.wikipedia.org/wiki/Sorting_algorithm
<https://blog.boot.dev/golang/insertion-sort-golang/>
<https://www.tutorialspoint.com/bubble-sort-in-go-lang>
https://youtu.be/_cCMkGRbXRM?si=aSZt_UAQDkU1bybs
https://youtu.be/OWCl_LEhhvU?si=oe8sPmEtyN3bp4dm
<https://golang-id.org/blog/go-slices-usage-and-internals/>
<https://thelinuxcode.com/bubble-sort-explained-a-step-by-step-tutorial-for-beginners/>
<https://www.geeksforgeeks.org/bubble-sort-algorithm/>
https://youtu.be/_cCMkGRbXRM?si=N17Zo138he5dyRo8

<https://thelinuxcode.com/bubble-sort-explained-a-step-by-step-tutorial-for-beginners/>