Laporan Tubes Sistem Aplikasi Reservasi Hotel:

Matakuliah Analisis Komplesitas Algoritma

Anggota Kelompok:

Relingga Aditya	103022300107
Nizar Abdul Fattah	103022300103

Kode Dosen :

LDS

Daftar isi

Pendahuluan

Di era modern saat ini, sistem reservasi hotel menjadi salah satu aspek penting di bidang operasional hotel. Hal ini bermandaat karena memberikan kemudahan bagi pelanggan. Dengan meningkatnya permintaan layanan yang cepat dan efisien, diperlukan solusi teknologi yang optimal untuk mengelola kamar dan statusnya. Dalam laporan ini, akan dianalisis dua pendekatan algoritma, yaitu iteratif dan rekursif, yang digunakan untuk memecahkan permasalahan pencarian kamar kosong.

Deskripsi Studi Kasus Permasalahan

Sistem reservasi hotel adalah salah satu sistem yang penting dalam operasional hotel modern. Di dalam bidang perhotelan. Pengelolaan data kamar yang efisien sangat krusial untuk meningkatkan pengalaman pelanggan dan memastikan operasional yang optimal. Studi kasus ini berfokus pada pencarian kamar kosong di hotel yang memiliki 100 kamar.

- Setiap kamar memiliki empat atribut utama:
- Nomor Kamar: Identitas unik untuk setiap kamar.
- Tipe Kamar: Kategori kamar seperti Single, Double, atau Suite.
- Harga Kamar: Biaya permalam berdasarkan tipe kamar.
- Status Kamar: Menunjukkan apakah kamar kosong atau terisi.

Permasalahan utama yang dihadapi adalah bagaimana mencari kamar kosong secara cepat dan efisien saat data kamar bertambah banyak. Permasalahan ini sangatlah penting karena permintaan pelanggan sering kali membutuhkan waktu respons yang sangat singkat, terutama pada musim liburan atau periode permintaan tinggi.Untuk mengatasi masalahi tersebut, digunakan dua pendekatan algoritma:

- Iteratif: Algoritma yang menggunakan perulangan untuk memeriksa status kamar.
- Rekursif: Algoritma yang memanfaatkan pemanggilan fungsi berulang untuk menyelesaikan pencarian secara bertahap.

Dengan membandingkan kedua algoritma ini, kita dapat menentukan metode mana yang lebih efisien dalam menangani kebutuhan operasional hotel, terutama dalam hal kompleksitas waktu dan memori. Studi ini juga mencakup analisis kinerja algoritma pada dataset yang berbeda ukuran untuk mendapatkan pemahaman yang lebih mendalam tentang kelebihan dan kekurangannya.

Sistem reservasi hotel adalah salah satu komponen penting dalam operasional hotel modern. Studi kasus ini berfokus pada pencarian kamar kosong di hotel dengan 100 kamar. Setiap kamar memiliki atribut seperti nomor kamar, tipe kamar, harga kamar, dan status (kosong atau terisi). Tantangan utama adalah mencari kamar kosong secara efisien menggunakan dua pendekatan algoritma: iteratif dan rekursif.

A.Deskripsi Dua Algoritma yang Dipilih untuk Menyelesaikan Permasalahan

a. Algoritma Iteratif

Pada algoritma iteratif, pencarian kamar kosong dilakukan dengan menggunakan perulangan (loop). Algoritma ini memeriksa setiap kamar secara berurutan untuk menentukan apakah status kamar adalah "kosong". Jika ditemukan kamar kosong, data kamar tersebut ditambahkan ke dalam daftar hasil.

Code menggunakan bahasa pemrogaman python:

```python
# Algoritma iteratif untuk mencari kamar kosong
def cari_kamar_kosong_iteratif(kamar):
    return [k for k in kamar if k['status'] == "Kosong"]
```

algoritma iteratif fungsi pencarian kamar kosong dilakukan dengan memanfaatkan perulangan (loop). Algoritma ini dirancang untuk memeriksa setiap kamar dalam daftar kamar secara berurutan. Setiap iterasi mengevaluasi atribut "status" dari kamar untuk menentukan apakah statusnya "kosong". Jika status kamar "kosong", maka kamar tersebut ditambahkan ke dalam daftar hasil, yang kemudian dikembalikan sebagai keluaran dari algoritma.

Proses pencarian dimulai dari kamar pertama hingga kamar terakhir dalam daftar kamar. Algoritma ini memanfaatkan struktur loop yang sederhana,Sehingga sangat intuitif dan mudah diimplementasikan. Dalam situasi di mana data kamar sangat besar, algoritma iteratif menawarkan stabilitas karena tidak bergantung pada tumpukan rekursi, sehingga aman dari batasan kedalaman rekursi.

Selain itu, algoritma ini tidak memerlukan alokasi memori tambahan yang signifikan, membuatnya lebih efisien dalam hal penggunaan memori dibandingkan dengan pendekatan rekursif. Keunggulan lainnya adalah algoritma iteratif mudah dimodifikasi untuk menambahkan logika pencarian yang lebih kompleks jika diperlukan.

Keuntungan:

- Sederhana untuk diimplementasikan.
- Tidak bergantung pada tumpukan rekursi, sehingga lebih aman terhadap batasan memori.

Kelemahan:

- Kurang elegan untuk masalah yang memerlukan pemrosesan rekursif alami.

b. Algoritma Rekursif

Algoritma rekursif berkerja dengan cara melakukan pencarian dengan memanggil fungsi secara berulang dengan parameter yang diperbarui hingga semua kamar diperiksa. Hasilnya disimpan dalam daftar yang diperbarui pada setiap panggilan fungsi.

Code menggunakan bahasa pemrogaman python:

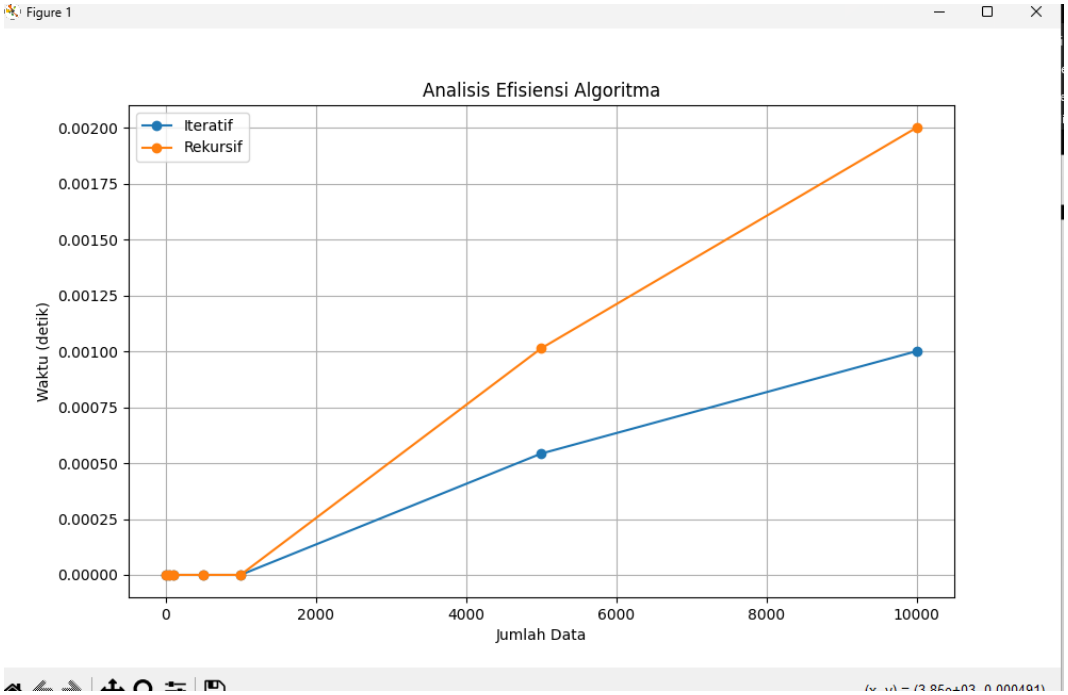
```python
# Algoritma rekursif untuk mencari kamar kosong
def cari_kamar_kosong_rekursif(kamar, index=0, hasil=None):
    if hasil is None:
        hasil = []
    if index >= len(kamar):
        return hasil
    if kamar[index]['status'] == "Kosong":
        hasil.append(kamar[index])
    return cari_kamar_kosong_rekursif(kamar, index + 1, hasil)
```

Keunggulan:

- Elegan untuk masalah yang dapat dipecah menjadi submasalah lebih kecil.

Kelemahan:

- Rentan terhadap batasan kedalaman rekursi.
- Membutuhkan lebih banyak memori dibandingkan iterasi.



B.Grafik perbandingan runtime

A. Algoritma Iteratif (Garis Biru)

Algoritma iteratif menempuh waktu eksekusi meningkat secara linear seiring bertambahnya jumlah data. Hal ini bisa terjadi karena algoritma iteratif menggunakan perulangan sederhana untuk memproses setiap elemen satu per satu. Dengan kompleksitas waktu O(n), algoritma ini menunjukkan stabilitas runtime yang konsisten. Tidak adanya penggunaan tumpukan rekursi membuat algoritma iteratif lebih efisien dalam hal memori dan waktu eksekusi, terutama pada dataset besar. Oleh karena itu, algoritma iteratif sangat cocok untuk kebutuhan skala besar dengan data yang banyak.

B.Algoritma Rekursif (Garis Oranye)

Algoritma rekursif memiliki performa yang lebih cepat untuk dataset kecil, karena overhead dari pemanggilan fungsi rekursi relatif kecil pada ukuran data yang terbatas. Namun, runtime algoritma rekursif meningkat tajam seiring bertambahnya jumlah data karena adanya tumpukan rekursi yang memerlukan memori tambahan pada setiap pemanggilan fungsi. Kompleksitas waktu algoritma rekursif tetap O(n), tetapi overhead ini membuatnya kurang efisien dibandingkan iteratif ketika jumlah data besar.

C. Kesimpulan
Kedua algoritma memiliki kelebihan masing-masing. Algoritma iteratif lebih efisien dan stabil untuk dataset besar karena tidak memiliki overhead tambahan dari tumpukan rekursi. Sementara itu, algoritma rekursif menawarkan solusi yang lebih elegan dan mudah dipahami untuk dataset kecil, tetapi kurang cocok untuk data dalam jumlah besar karena penggunaan memori tambahan. Oleh karena itu, pemilihan algoritma sangat bergantung pada ukuran dataset dan kebutuhan efisiensi implementasi.

C.Analisis Perbandingan Kedua Algoritma

1. Kompleksitas Waktu:
Algoritma iteratif maupun Rekursif memiliki kompleksitas waktu O(n), di mana n adalah jumlah kamar yang diperiksa. Namun, dalam implementasi praktis, algoritma rekursif membutuhkan waktu tambahan karena adanya overhead dari pemanggilan fungsi secara berulang. Sebaliknya, algoritma iteratif memanfaatkan perulangan sederhana, sehingga runtime-nya lebih konsisten dan tidak terpengaruh oleh overhead tambahan.

2. Kompleksitas Memori:
Algoritma iteratif menggunakan ruang memori tetap (constant space), karena tidak memerlukan alokasi memori tambahan untuk setiap iterasi. Sementara itu, algoritma rekursif menggunakan ruang memori O(n) untuk tumpukan rekursi, di mana setiap pemanggilan fungsi menambah penggunaan memori. Hal ini membuat algoritma rekursif menjadi kurang efisien pada dataset besar, terutama jika jumlah data mendekati batas maksimum kedalaman rekursi.

3. Efektivitas Implementasi:
Dari segi implementasi, algoritma rekursif lebih elegan dan intuitif untuk masalah yang dapat dipecah menjadi submasalah kecil, seperti pencarian kamar kosong. Namun, algoritma iteratif lebih mudah dioptimalkan untuk menangani dataset besar, karena stabilitasnya terhadap batasan memori.

4. Performansi pada Dataset Kecil dan Besar:

- Dataset Kecil: Algoritma rekursif dapat bersaing dengan iteratif dalam hal runtime dan efisiensi memori karena overhead pemanggilan fungsi tidak signifikan.
- Dataset Besar: Algoritma iteratif menjadi pilihan yang lebih unggul karena runtime yang lebih stabil dan penggunaan memori yang lebih rendah.

Refrensi

https://www.python.org/doc/

https://docs.python.org/3/library/tkinter.html

https://docs.python.org/3/library/math.htm

https://www.geeksforgeeks.org/difference-between-recursion-and-iteration/