

SISTEM INFORMASI DAN ANALISIS OBAT DENGAN BENCHMARKING

Pengelolaan informasi obat yang akurat dan terintegrasi sangat penting dalam layanan kesehatan untuk mendukung pengambilan keputusan medis, memastikan keamanan pasien, dan meningkatkan efisiensi operasional. Namun, banyak sistem informasi obat saat ini menghadapi masalah seperti kurangnya integrasi data, informasi yang tidak konsisten, dan kesulitan menilai kualitas obat, yang dapat menyebabkan kesalahan pemberian obat dan pemborosan biaya.

```
# Fungsi iteratif untuk pencarian obat
def cari_obat_iteratif(keyword, data):
    """
    Pencarian obat secara iteratif.
    keyword : string yang dicari.
    data : list dictionary data obat.
    """
    hasil = []
    n = len(data)

    for i in range(n):
        if keyword.lower() in data[i]["nama"].lower():
            hasil.append(data[i])

    return hasil
```

```
# Fungsi rekursif untuk pencarian obat
def cari_obat_rekursif(keyword, data, index=0, hasil=None):
    """
    Pencarian obat secara rekursif.
    keyword : string yang dicari.
    data : list dictionary data obat.
    index : indeks saat ini dalam pencarian.
    hasil : list untuk menyimpan hasil pencarian.
    """
    if hasil is None:
        hasil = []

    # Kondisi basis: jika indeks melebihi panjang data
    if index >= len(data):
        return hasil

    # Cek apakah keyword ditemukan
    if keyword.lower() in data[index]["nama"].lower():
        hasil.append(data[index])

    # Rekursif dengan indeks berikutnya
    return cari_obat_rekursif(keyword, data, index + 1, hasil)
```

ITERATIF

Penjelasan Program Iteratif
Fungsi: cari_obat_iteratif(keyword, data)
1.Tujuan: Fungsi ini mencari data obat berdasarkan keyword secara iteratif.
2.Parameter:

- keyword: String yang ingin dicari.
- data: List berisi dictionary data obat.

3.Proses:

- Inisialisasi: Membuat list kosong hasil untuk menyimpan data obat yang sesuai.
- Iterasi:
 - Menggunakan loop for untuk memeriksa setiap item dalam data.
 - Jika keyword ditemukan dalam nama obat (setelah diubah ke huruf kecil), data tersebut ditambahkan ke list hasil.

4.Pengembalian: Mengembalikan list hasil berisi semua obat yang sesuai.

REKURSIF

Penjelasan Program Rekursif
Fungsi: cari_obat_rekursif(keyword, data, index=0, hasil=None)
1.Tujuan: Fungsi ini mencari data obat berdasarkan keyword secara rekursif.
2.Parameter:

- keyword: String yang ingin dicari.
- data: List berisi dictionary data obat.
- index: Indeks saat ini dalam pencarian (default 0).
- hasil: List untuk menyimpan hasil pencarian (default None).

3.Proses:

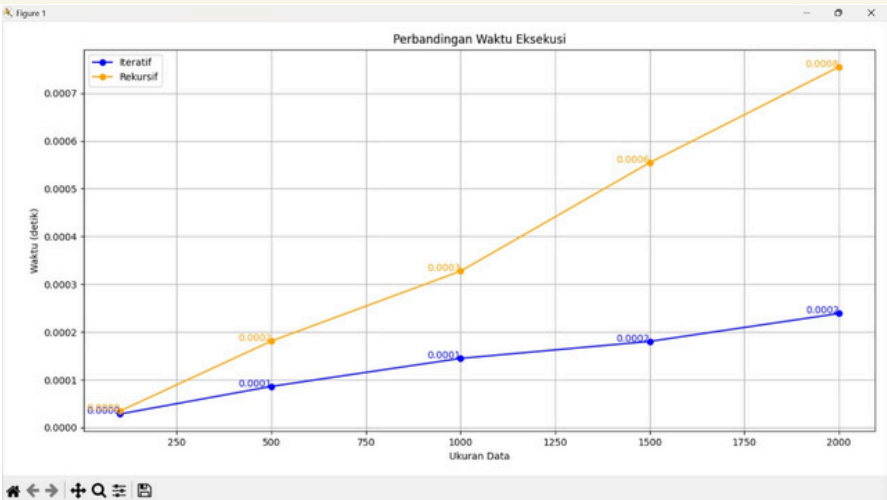
- Inisialisasi: Jika hasil belum diinisialisasi (None), maka diubah menjadi list kosong.
- Kondisi Basis: Fungsi akan berhenti jika indeks sudah melebihi panjang list data.
- Pencocokan:
 - Nama obat pada indeks saat ini (data[index]["nama"]) diubah ke huruf kecil.
 - Jika keyword ada di nama obat, maka dictionary obat tersebut ditambahkan ke hasil.
- Panggilan Rekursif: Fungsi memanggil dirinya sendiri dengan index bertambah 1 untuk memproses data berikutnya.

4.Pengembalian: Mengembalikan list hasil berisi semua obat yang sesuai.

ANALYSIS

Diagram menunjukkan bahwa algoritma iteratif lebih efisien dibandingkan algoritma rekursif dalam hal waktu eksekusi, terutama untuk data berukuran besar. Pada ukuran data kecil, perbedaan waktu eksekusi antara keduanya tidak terlalu signifikan.

Namun, saat ukuran data meningkat, waktu eksekusi algoritma rekursif bertambah jauh lebih cepat dibandingkan iteratif, menunjukkan bahwa algoritma rekursif memiliki kompleksitas waktu yang lebih tinggi. Oleh karena itu, algoritma iteratif lebih cocok digunakan untuk kasus yang melibatkan data berukuran besar karena lebih stabil dan efisien.



CONCLUSION

- Tujuan: Kedua fungsi mencari data obat berdasarkan keyword di field nama pada list data.
- Pendekatan:
 - Rekursif: Memanggil fungsi secara berulang hingga kondisi basis tercapai. Elegan tapi memakan memori lebih besar.
 - Iteratif: Menggunakan loop untuk iterasi, lebih efisien untuk dataset besar.
- Persamaan: Hasil output sama, tidak case-sensitive, dan cocok untuk pencarian berbasis string.
- Rekomendasi:
 - Gunakan rekursif untuk dataset kecil.
 - Gunakan iteratif untuk dataset besar dan efisiensi memori.

