

# MENCARI RATING FILM TERBAIK MARVEL DAN DC DI IMDB MENGGUNAKAN SEQUENTIAL SEARCH

Naufal Muhammad Dzulfikar (103022300021)

Frizam Dafa Maulana (103022300011)

## Studi Kasus

Dataset yang disediakan berisi kumpulan film-film yang dirilis oleh dua studio besar, yaitu Marvel dan DC. Setiap entri dalam dataset tersebut mencakup berbagai informasi penting, seperti judul film, tahun rilis, serta rating IMDb dari masing-masing film. Data ini memungkinkan kita untuk melakukan analisis lebih mendalam, termasuk mencari film dengan rating tertinggi berdasarkan kriteria tertentu.

Dalam skenario ini, pengguna diberikan opsi untuk memilih apakah mereka ingin mencari film dari Marvel atau DC. Berdasarkan pilihan tersebut, sistem akan memfilter data untuk hanya menyertakan film-film yang berasal dari studio yang dipilih. Setelah data difilter, langkah selanjutnya adalah mencari film dengan rating tertinggi dalam kelompok tersebut. Jika terdapat lebih dari satu film dengan rating tertinggi yang sama, maka sistem akan menampilkan salah satu dari film-film tersebut.

## Tujuan

Tujuan utama dari pembuatan tugas besar ini adalah untuk melakukan analisis dan implementasi algoritma dalam mencari rating IMDB terbaik dari film-film yang diproduksi oleh Marvel atau DC. Pencarian ini akan dilakukan dengan menggunakan algoritma sequential search dalam dua bentuk, yaitu secara iteratif dan rekursif.

## Kesimpulan

Algoritma sequential search dapat digunakan untuk menemukan film dengan rating IMDb tertinggi berdasarkan kategori studio, yaitu Marvel atau DC. Dua metode algoritma sequential search yang digunakan adalah pendekatan iteratif dan rekursif. Pendekatan iteratif menggunakan perulangan untuk memeriksa setiap elemen dalam dataset dan memperbarui film dengan rating tertinggi berdasarkan kriteria tertentu. Sementara itu, pendekatan rekursif menggunakan pemanggilan fungsi secara berulang untuk mencapai hasil yang sama, dengan memanfaatkan kondisi dasar untuk mengakhiri rekursi. Analisis ini menunjukkan bagaimana algoritma sederhana dapat diimplementasikan secara efektif untuk menyelesaikan permasalahan pencarian dalam dataset dengan ukuran kecil hingga sedang.

## Algoritma Sequential Search Iteratif

```
public static Film findBestRatingIterative(Film[] films, String category) {  
    Film bestFilm = null;  
    double maxRating = 0.0;  
    int n = films.length;  
  
    for (int i = 0; i < n; i++) {  
        if (films[i].getStudio().equalsIgnoreCase(category) && films[i].getImdbRating() > maxRating) {  
            maxRating = films[i].getImdbRating();  
            bestFilm = films[i];  
        }  
    }  
  
    return bestFilm;  
}
```

## Algoritma Sequential Search Rekursif

```
public static Film findBestRatingRecursive(Film[] films, String category, int n, Film currentBest) {  
    // Base case: jika n = 0  
    if (n == 0) {  
        return currentBest;  
    } else if (films[n - 1].getStudio().equalsIgnoreCase(category) &&  
        (currentBest == null || films[n - 1].getImdbRating() > currentBest.getImdbRating())) { // Periksa elemen ke-(n-1)  
        currentBest = films[n - 1];  
    }  
    // Recursive case: lanjut ke elemen sebelumnya  
    return findBestRatingRecursive(films, category, n - 1, currentBest);  
}
```

## Perhitungan Kompleksitas Algoritma Iteratif

$$\sum_{i=0}^n 1 = n + 0 + 1 = n + 1 \in O(n)$$

## Perhitungan Kompleksitas Algoritma Rekursif

$$T(n) \begin{cases} 0, & n = 0 \\ T(n - 1) + 1, & n > 0 \end{cases}$$

Mencari Persamaan Karakteristik

$$(r - 1)(r - 1) = 0$$
$$r_1 = r_2 = 1$$

Solusi umum

$$T(n) = C_1(r_1^n) + C_2 n(r_2^n) = C_1(1^n) + C_2 n(1^n) = C_1 + C_2 n$$

Nilai T(1)

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

Nilai konstanta C<sub>1</sub> dan C<sub>2</sub>

$$C_1 = 0 \text{ \& } C_2 = 1$$

Solusi khusus

$$T(n) = C_1 + C_2 n$$

$$T(n) = 0 + 1(n) \in O(n) \text{ (elemen Big-O(n))}$$

