

Algoritma dan Struktur Data

Algoritma Pencarian (*Searching Algorithm*)

Umi Sa'adah

Tita Karlita

Entin Martiana Kusumaningtyas

Arna Fariza

2021



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Materi

- Pengenalan algoritma pencarian
- Algoritma *sequential search*
- Algoritma *binary search*

Pengenalan algoritma pencarian

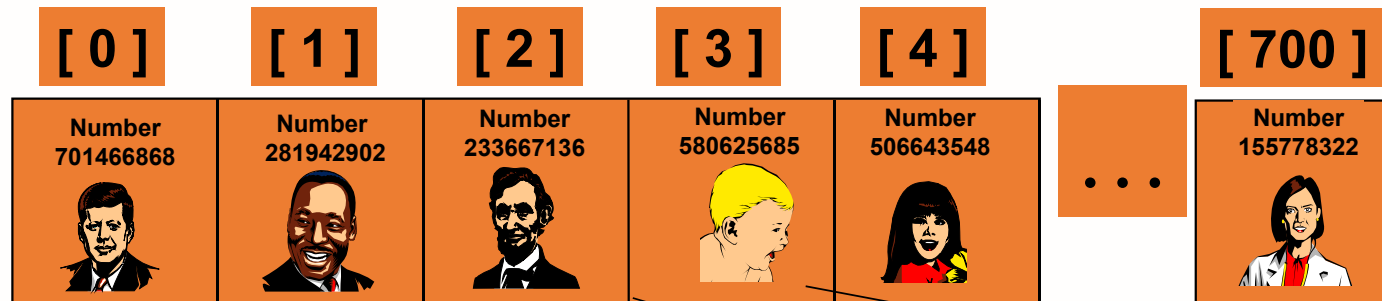
- Algoritma yang digunakan untuk mencari sebuah data dan lokasinya pada kumpulan data yang telah ada.
- Data yang dicari disebut dengan **kata kunci / key**.
- Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari:
 - ditemukan (*successful*)
 - tidak ditemukan (*unsuccessful*)



Find
Key = "Surabaya"



Searching Concepts



Setiap record dalam list memiliki sebuah associated key.
Pada contoh ini, key-nya = ID numbers.

Diberikan sebuah key, bagaimana cara menemukan recordnya dari list secara efektif ?

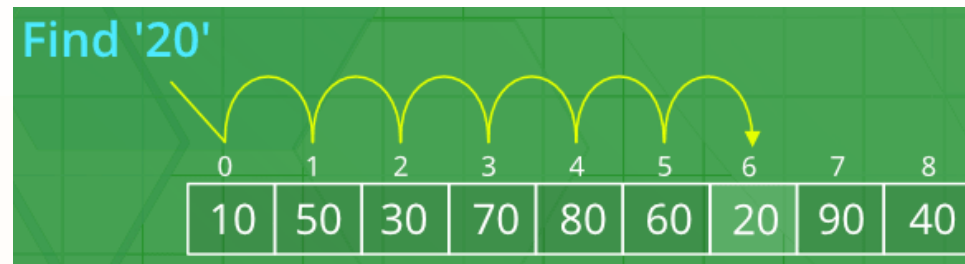


Algoritma pencarian

- Ada dua macam teknik pencarian yaitu:
 - pencarian sekuensial (*sequential search*)
 - pencarian biner (*binary search*)
- Perbedaan dari dua teknik ini terletak pada keadaan data.
 - Pencarian sekuensial digunakan apabila data dalam keadaan acak (tidak terurut) atau data yang sudah terurut.
 - Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

Sequential Search

- Dimulai dari data awal.
- Cek seluruh data dalam array, baca satu persatu.
- Temukan data yang sesuai dengan key yang dicari.
- Proses searching berhenti karena salah satu alasan berikut:
 - Sukses – key ditemukan
 - End of List – Sudah tidak ada data yang dibandingkan
- Diaplikasikan pada data (sorted & unsorted)

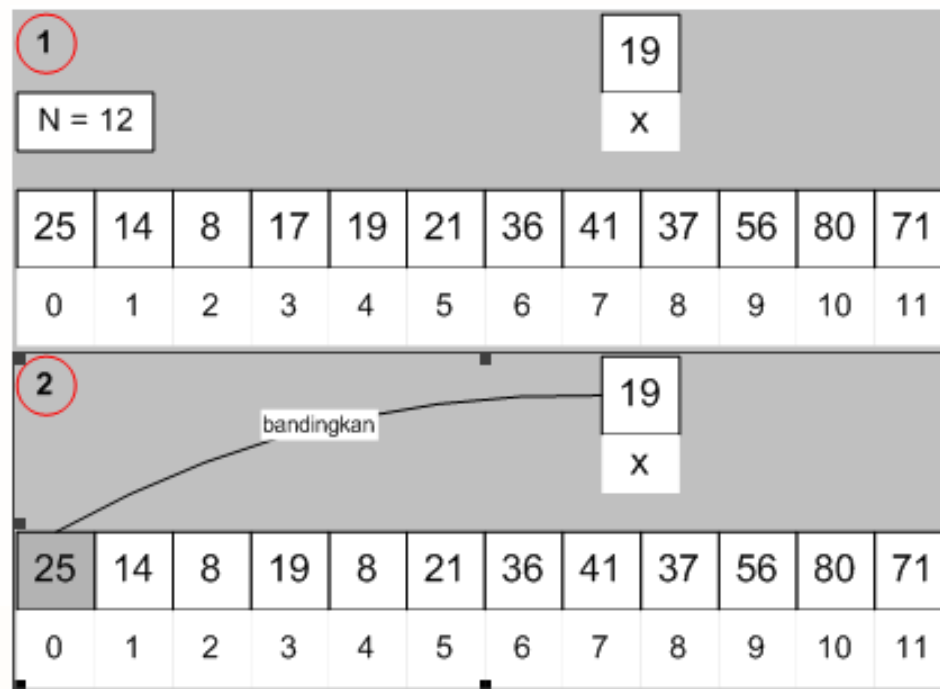


Algoritma *Sequential Search*

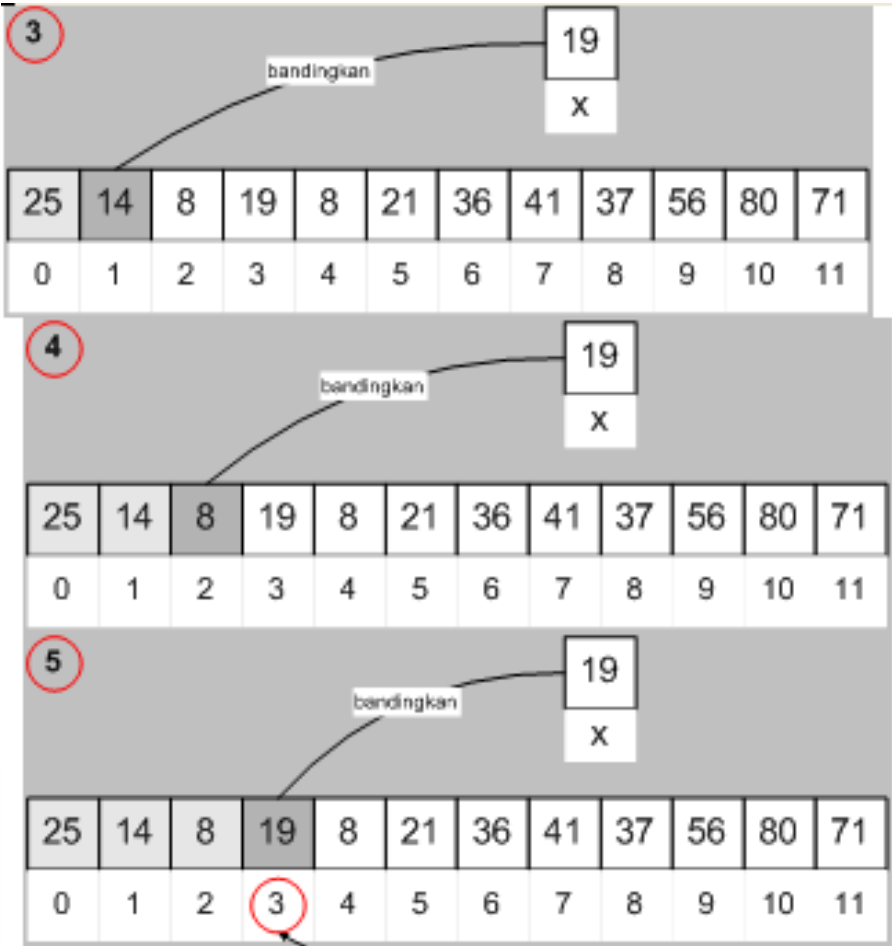
1. $i \leftarrow 0$
2. $\text{ketemu} \leftarrow \text{false}$
3. Selama (tidak ketemu) dan $(i < N)$ kerjakan baris 4
4. Jika $(\text{Data}[i] == \text{key})$ maka
 $\text{ketemu} \leftarrow \text{true}$
 jika tidak
 $i \leftarrow i + 1$
5. Jika (ketemu) maka
 i adalah indeks dari data yang dicari
 jika tidak
 data tidak ditemukan



Ilustrasi *Sequential Search*



Ilustrasi
Sequential Search



19 ketemu pada indeks ke 3 ←



Sequential Search Analysis

- Jumlah operasi pencarian bergantung pada nilai n , yaitu jumlah elemen dalam list

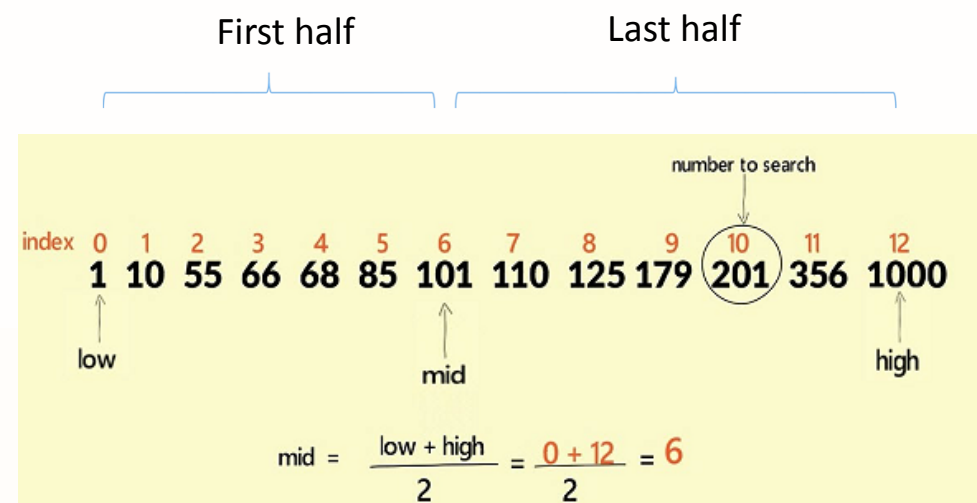
Worst Case Time for Sequential Search

- Kondisi yang mengharuskan pengecekan terhadap semua elemen array (n data):
 - Data yang dicari berada pada posisi terakhir dari array
 - Setelah pengecekan seluruh elemen array, ternyata record yang dicari tidak berhasil ditemukan dalam array tersebut
- Untuk sebuah array dengan n elemen, maka worst case time untuk sequential search → membutuhkan n kali akses data: $O(n)$.



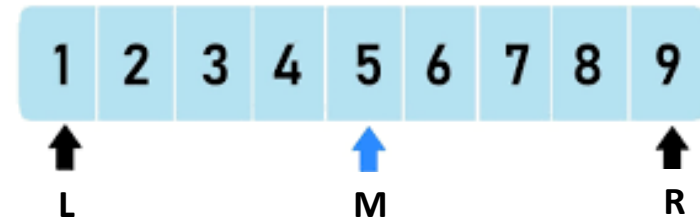
Binary Search

- Untuk sejumlah N data yang telah terurut
- Ulangi sampai selesai
 - Pilih data yang terletak di tengah (middle element)
 - Bandingkan nilai key dengan key terpilih (middle element)
 - Hasil perbandingan:
 - Key < middle element: Range = First half
 - Key > middle element : Range = Last half
 - Key = middle element : Success, Done



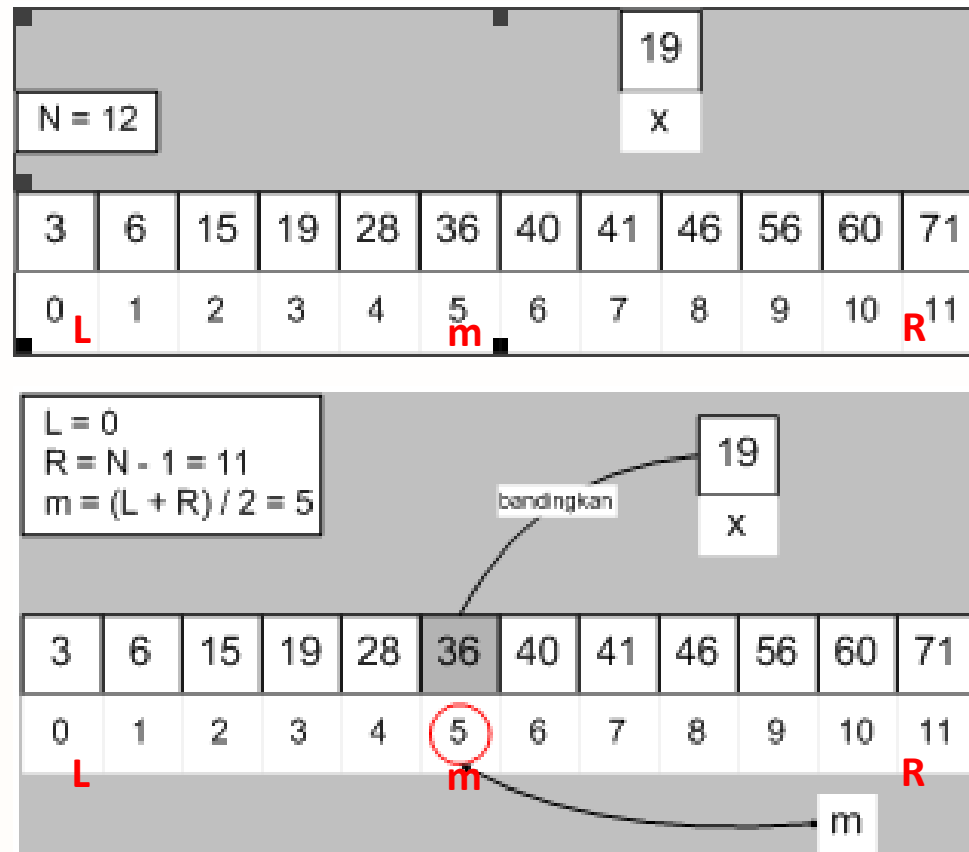
Pseudocode Binary Search

1. $L \leftarrow 0$
2. $R \leftarrow N - 1$
3. `ketemu` \leftarrow `false`
4. Selama ($L \leq R$) dan (`tidak ketemu`) kerjakan baris 5 sampai dengan 8
5. $m \leftarrow (L + R) / 2$
6. Jika (`Data[m] == key`) maka `ketemu` \leftarrow `true`
7. Jika (`key < Data[m]`) maka $R \leftarrow m - 1$
8. Jika (`key > Data[m]`) maka $L \leftarrow m + 1$
9. Jika `ketemu`
 maka m adalah indeks dari data yang dicari,
 jika `tidak ketemu`
 data tidak ditemukan



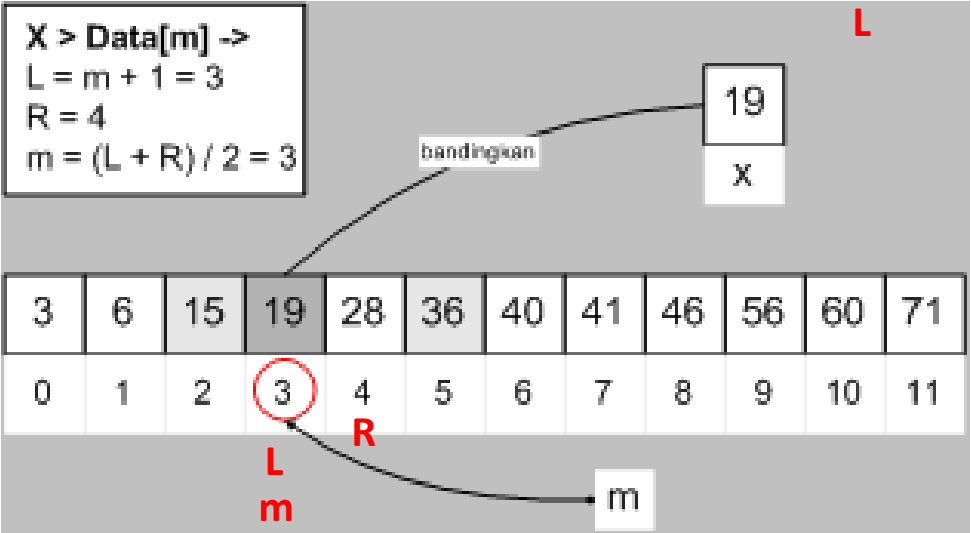
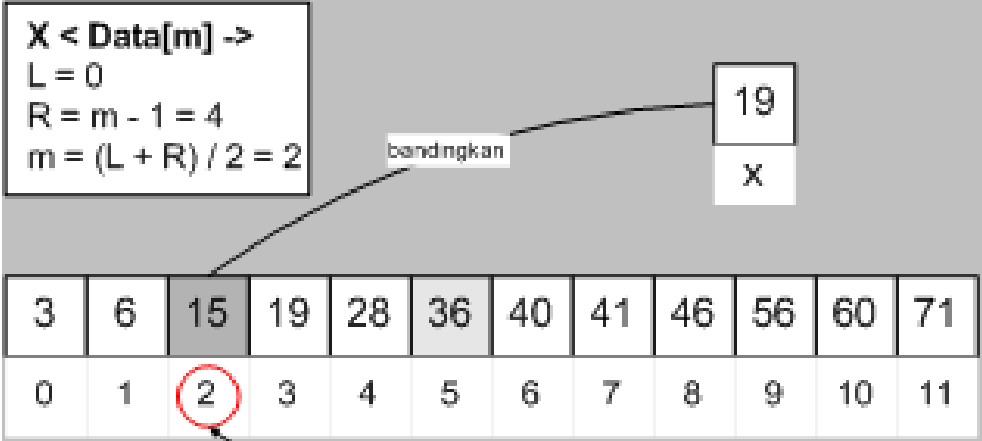
Binary Search

- Studi Kasus 1



Binary Search

- Studi Kasus 1



Binary Search

- Studi Kasus 2

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

Binary Search

Key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

$$m = (0+6) / 2 = 3$$

Binary Search

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Apakah $key == m$? NO.

Binary Search

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53




Apakah $key < m$? YES.

Binary Search

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Pencarian key pada area sebelum m

Binary Search

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

$$m = (0+2) / 2 = 1$$



Binary Search

Pencarian data dengan $key=7$

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Apakah $key == m$? NO.

Binary Search

Pencarian data dengan $key=7$

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Apakah $key < m$? NO.

Binary Search

Pencarian data dengan $key=7$

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53




Apakah $key > m$? YES.

Binary Search

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Pencarian target pada area setelah m

Binary Search

Pencarian data dengan key=7

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Apakah key = m ? YES.

Searching algorithm: Time complexity

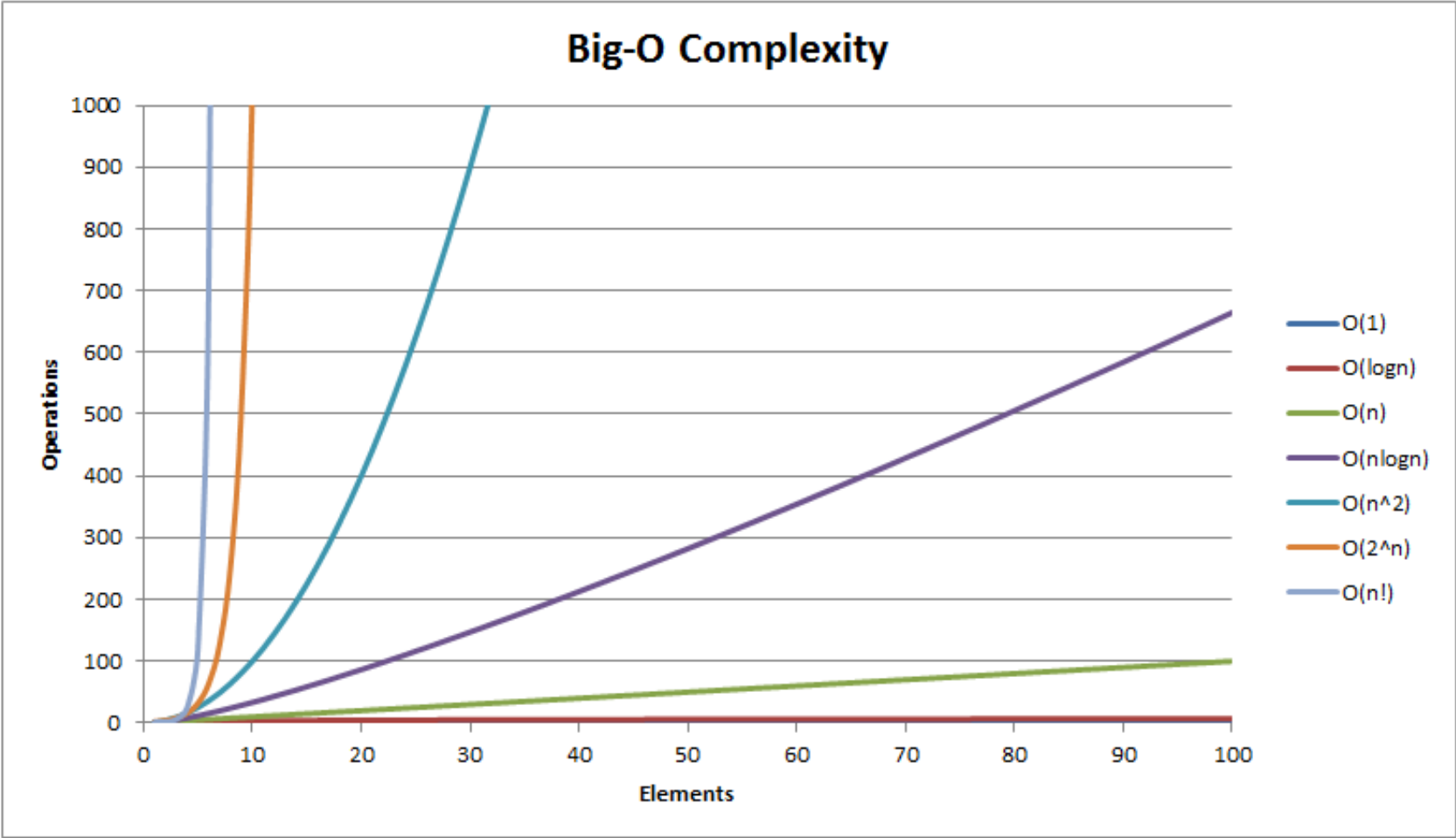
	Time Complexity
Linear Search	$O(n)$
Binary Search	$O(\log(n))$
Jump Search	$O(\sqrt{n})$
Interpolation Search	$O(\log(\log n))$ -Best $O(n)$ -Worst
Exponential Search	$O(\log(n))$
Sequential search	$O(n)$
Depth-first search (DFS)	$O(V + E)$
Breadth-first search (BFS)	$O(V + E)$



Searching algorithm: Time complexity

Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	$O(1)$	$O(n)$	$O(n)$	$O(1)$
Binary Search	$O(1)$	$O(\log n)$	$O(\log n)$	$O(1)$
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Quick Sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$
Heap Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Bucket Sort	$O(n+k)$	$O(n+k)$	$O(n^2)$	$O(n)$
Radix Sort	$O(nk)$	$O(nk)$	$O(nk)$	$O(n+k)$
Tim Sort	$O(n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Shell Sort	$O(n)$	$O((\log(n))^2)$	$O((\log(n))^2)$	$O(1)$





Kesimpulan

- Ada dua macam teknik pencarian yaitu pencarian:
 - sekuensial (*sequential search*)
 - pencarian biner (*binary search*)
- Algoritma yang digunakan untuk mencari lokasi dari sebuah data yang diberikan (disebut kata kunci) pada kumpulan data yang telah ada.
- Perbedaan dari dua teknik ini terletak pada keadaan data.
 - Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak terurut.
 - Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.
- Kompleksitas waktu
 - Sequential search: $O(n)$
 - Binary search: $O(\log n)$



Latihan Soal

- Dengan Pencarian Sekuensial, carilah data 9 dari data berikut
- 10 1 2 11 8 7 5 4 15 16
- Dengan Pencarian Biner, carilah data 9 dari data berikut
- 2 5 6 9 10 12 15 18 20 22