

# ANALISIS ALGORITMA

Week 04: Batas Bawah Kompleksitas Problem Komputasi

PROGRAM PASCA SARJANA INFORMATIKA  
FAKULTAS TEKNIK INFORMATIKA  
UNIVERSITAS TELKOM

2022/2023

# Batas Bawah Kompleksitas

bagian 3 dari 2

- Solusi kasus khusus

---

Pengurutan dengan waktu Linear

# Counting Sort

Diciptakan oleh H.H. Seward (1954)



1. Hitung/Akumulasi kemunculan setiap elemen data
  - Data boleh duplikat
2. Buat informasi kumulatif dari hasil akumulasi
  - yaitu  $kum(j) = \sum_{i=terkecil..j} acc(i)$ ;  $j = terkecil+1 .. terbesar$
3. Elemen data  $v$  berada pada posisi terurut di  $kum(v)$ 
  - Mulai dari elemen data paling akhir, agar hasil pengurutan stabil
  - Setelah penempatan decrement  $kum(v)$ , untuk posisi elemen sebelumnya

# Counting Sort

1	2	3	4	5	6	7	8	9
1	9	4	5	8	1	7	6	8

data awal

# Counting Sort

```
proc CountingSort( arr[1..n] )  
  acc[terkecil..terbesar] = 0  
  for i = 1 to n do  
    acc[arr[i]]++  
  endfor  
  for j = terkecil+1 to terbesar do  
    acc[j] := acc[j] + acc[j-1]  
  endfor  
  for i = n downto 1 do  
    dest[acc[arr[i]]] := arr[i]  
    acc[arr[i]]--  
  endfor  
  arr := dest
```

# Kompleksitas Counting Sort

- Untuk rentang data  $k = \text{terbesar} - \text{terkecil} + 1$ ,
  - Inisialisasi akumulator  $\text{acc} [ ]$  perlu  $O(k)$
- Dengan data sebanyak  $n$ 
  - Pencacahan isi  $\text{arr} [ ]$  adalah  $O(n)$
- Akumulasi hasil pencacahan di  $\text{acc} [ ]$  adalah  $O(k)$
- Distribusi data dari  $\text{arr} [ ]$  melalui  $\text{acc} [ ]$  ke  $\text{dest} [ ]$  adalah  $O(n)$
- Total waktu adalah  $O(k+n)$
- Jika  $k = O(n)$  maka Countingsort adalah  $O(n)$

# Observasi Counting Sort

- Kinerja Counting sort **bergantung rentang data** yang akan diurutkan
- Counting sort bersifat stabil



# Metoda Sorting Waktu Linear Lain

- Radix sort
  - Urut berdasarkan (kelompok) digit, dimulai digit kecil
  - Perkelompok digunakan sorting stabil, misalnya counting sort
  - Digunakan untuk mengurutkan kartu (mekanik)
  - Kompleksitas bergantung pada jumlah (kelompok) digit
- Bucket sort
  - Bagi data dalam beberapa kelompok
  - Ekspektasi jumlah data dalam satu kelompok/bucket adalah kecil (konstan)
  - Gunakan sorting biasa untuk setiap bucket, misalnya insertion sort
  - Kompleksitas bergantung pada realitas isi setiap bucket

# Radixsort

1	2	3	4	5	6	7	8	9
23	43	81	17	74	62	43	39	58

data awal

1	2	3	4	5	6	7	8	9
81	62	23	43	43	74	17	58	39

1 2 3 3 3 4 7 8 9

urutkan dari digit kanan

1	2	3	4	5	6	7	8	9
17	23	39	43	43	58	62	74	81

1 2 3 4 4 5 6 7 8

urutkan ke digit berikutnya  
selesai

Akhir dari Topik Minggu 04