POLITEKNIK STATISTIKA STIS

For Better Official Statistics

SORTING

Ibnu Santoso, Ratih Ngestrini, Nori Wilantika

PENGURUTAN DATA

- Pengurutan adalah proses menyusun data menurut aturan tertentu.
- Ingat pertemuan sebelumnya tentang pencarian (searching), jika data telah terurut maka proses pencarian dapat lebih efisien.

- Data yang diurutkan dapat berupa numerik ataupun karakter.
- Jenis pengurutan data:
  - Ascending (dari nilai terkecil ke terbesar) >> kita akan membahas ini
  - Descending (dari nilai terbesar ke terkecil)
- Untuk memudahkan pemahaman konsep, pada materi ini pengurutan akan dikhususkan pada data yang berstruktur array.

METODE PENGURUTAN DATA SEDERHANA

- **Bubble Sort**, pengurutan berdasar perbandingan
- **Selection Sort**, pengurutan berdasar prioritas/ seleksi
- **Insertion Sort**, pengurutan berdasarkan penyisipan dan penjagaan terurut

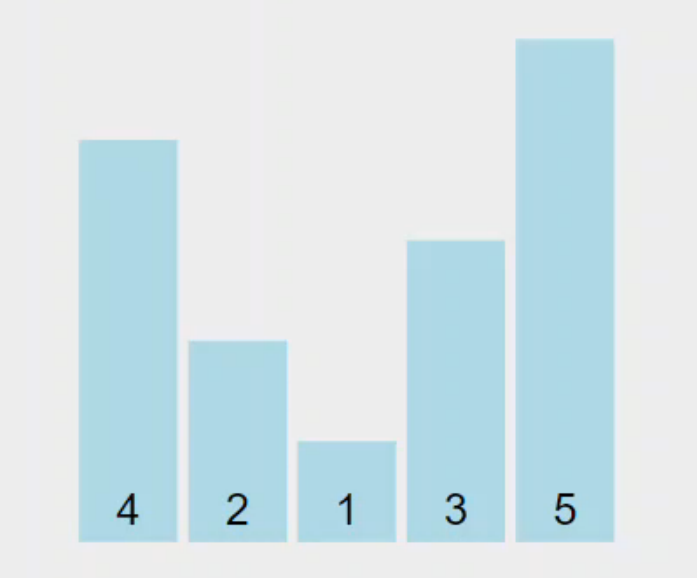
BUBBLE SORT

- Metode termudah.
- Bubble = busa/udara dalam air apa yang terjadi?
- Setiap proses membandingkan satu elemen dengan elemen berikutnya, seperti busa didalam air: yang ringan(busa) naik, yang berat(air) akan turun.

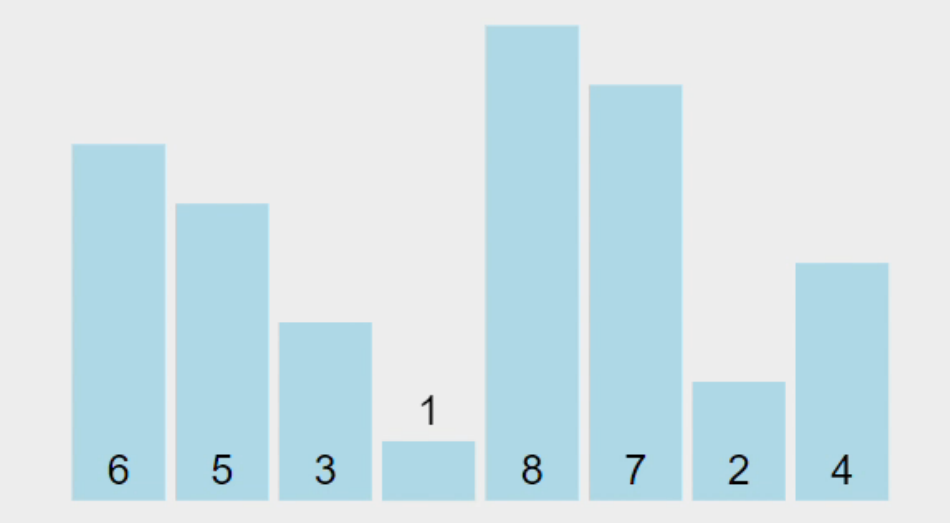
PROSES BUBBLE SORT

- Pengurutan data dilakukan dengan cara membandingkan masing-masing elemen, data sebelum dengan sesudahnya mana yang lebih besar atau kecil lalu ditukarkan **bila perlu**, secara terus menerus sampai data tersebut terurut.

- Ada beberapa cara
  - Pembandingan mulai dari depan
  - Pembandingan mulai dari belakang



4	2	1	3	5



6	5	3	1	8	7	2	4

IMPLEMENTASI PROGRAM BUBBLE SORT


```
Type Larik = array [1..100] of integer;
procedure Bubble(Arr:Larik; n:integer;);
var i,j: integer;
begin
        for i:= 1 to n-1 do begin
          for j:=1 to n-i do begin
              if Arr[j] > Arr[j+1] then
                  Tukar(Arr[j], Arr[j+1]);
          end;
        end;
end;
```

Jumlah Iterasi

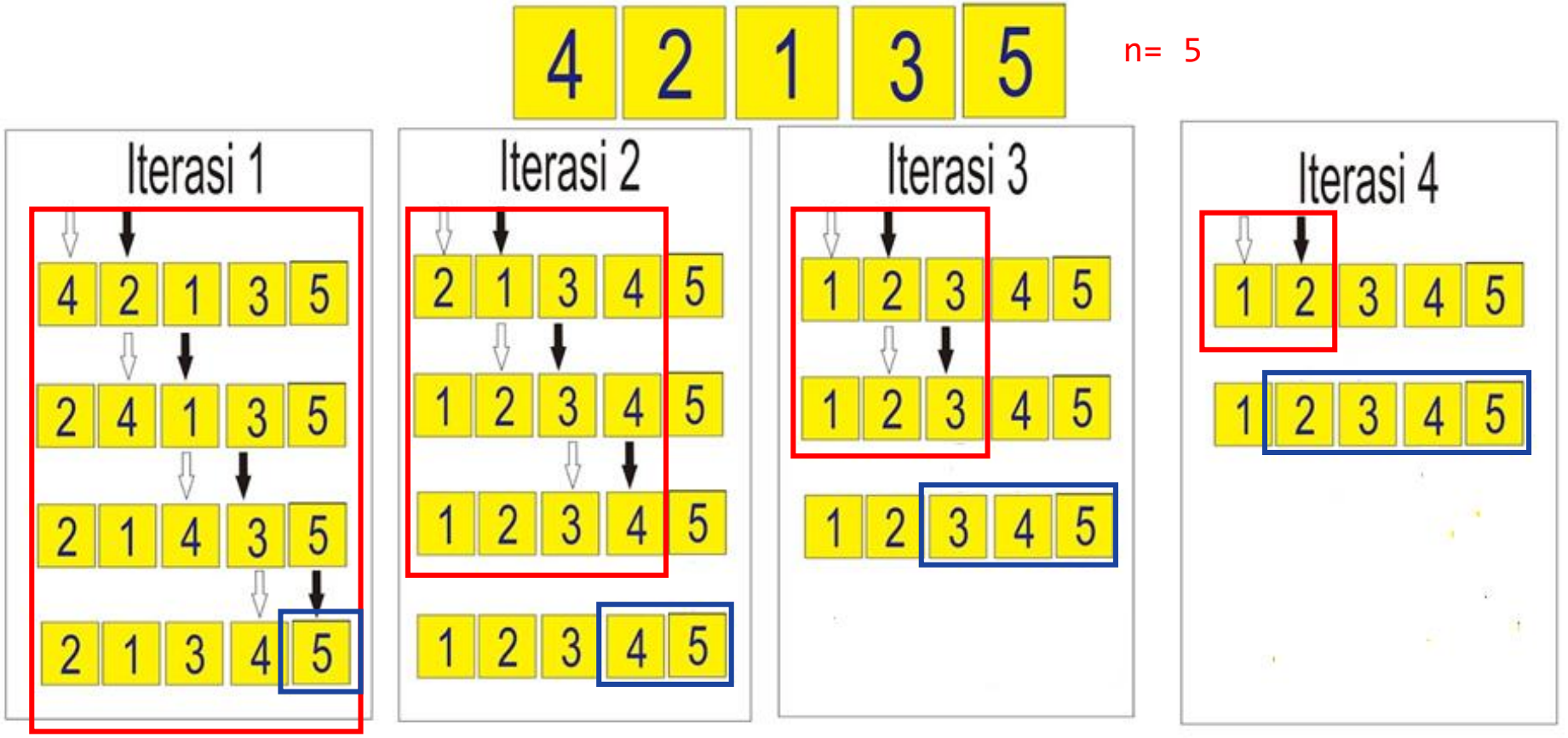
Perbandingan mulai dari depan

Membandingkan elemen dengan elemen sesudahnya

Iterasi elemen-elemen di dalam array hanya sampai n-i saja, karena elemen setelahnya pasti sudah urut.

Catatan: **Pembandingan bisa juga dimulai dari belakang**

- Perulangan pertama (jumlah data = n):
  - Proses 1: data ke **n** dibandingkan dgn data ke **n-1**, jika data ke n lebih kecil maka tukar data tsb,
  - Proses 2: data ke **n-1** dibandingkan dgn data ke **n-2**, jika data ke n lebih kecil maka tukar data tsb,
  - Demikian seterusnya sampai proses ke n-1.
- Perulangan ke 2:
  - Proses 1: data ke **n** dibandingkan dgn data ke **n-1**, jika data ke n lebih kecil maka tukar data tsb,
  - Proses 2: data ke **n-1** dibandingkan dgn data ke **n-2**, jika data ke n lebih kecil maka tukar data tsb,
  - Demikian seterusnya sampai proses ke n-2
- Perulangan ini dilakukan sebanyak n-1 kali



i= 1
j= 1 2 3 4

i= 2
j= 1 2 3

i= 3
j= 1 2

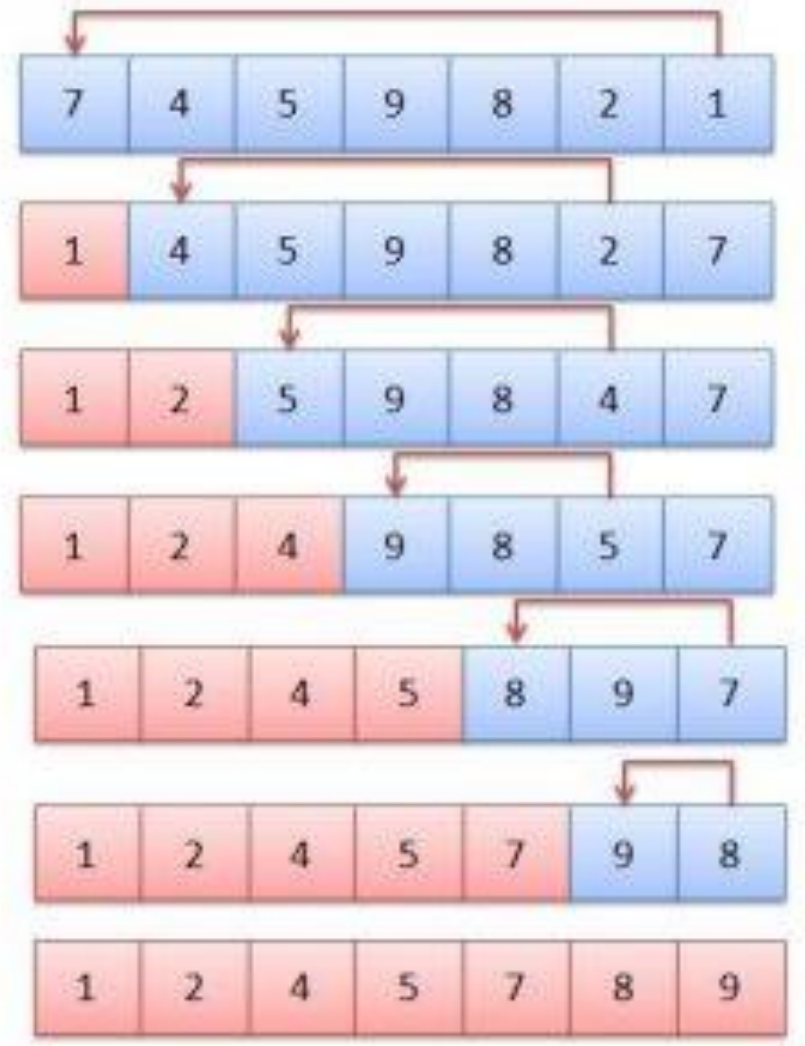
i= 4
j= 1

SELECTION SORT

- Kombinasi sorting dan searching
- Setiap perulangan, mencari elemen **terbesar/terkecil** pada bagian tertentu kemudian tukar dengan posisi yang sesuai
- Perulangan dilakukan sebanyak n-1 kali, dimana n adalah jumlah data

- Proses selection sort:
  - Perulangan 1:
    Cari nilai **minimum** dari data ke **1** sampai data ke n, tukar data tersebut dengan data ke **1**.
  - Perulangan 2:
    Cari nilai **minimum** dari data ke **2** sampai data ke n, tukar data tersebut dengan data ke **2**.
  - Dan seterusnya sampai perulangan ke n-1.

ILUSTRASI SELECTION SORT



Bagian yang sudah diurutkan

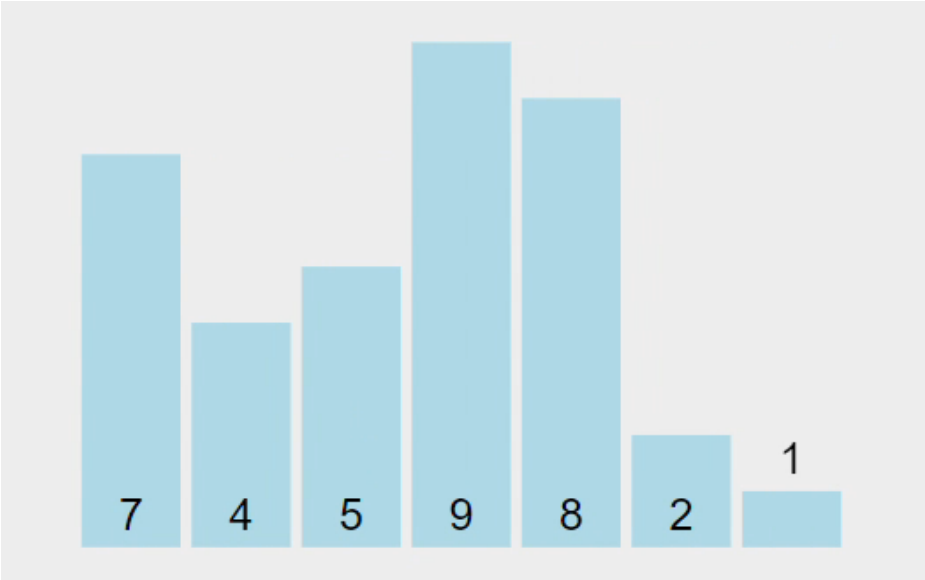
Bagian yang belum diurutkan

1. Langkah pertama dicari data terkecil dari data pertama sampai data terakhir. Kemudian data terkecil ditukar dengan data pertama. Dengan demikian, data pertama sekarang mempunyai nilai paling kecil dibanding data yang lain.

2. Langkah kedua, kita cari data terkecil mulai dari data kedua sampai terakhir. Data terkecil yang kita peroleh ditukar dengan data kedua

3. Demikian seterusnya sampai semua elemen dalam keadaan terurutkan.

Cara lain: yang dicari adalah nilai **maksimum** dan ditukar dengan data ke **n** dst.




```
Type Larik = array [1..100] of integer;
procedure Selection_Sort(Arr:Larik; n:integer;);
var i,j,idx_min: integer;
begin
    for i:= 1 to n-1 do begin
        begin
            idx_min = Arr[i];
            for j:= i+1 to n do
                if Arr[j] < Arr[idx_min] then idx_min:=j;
            Tukar(Arr[i],Arr[idx_min]);
        end;
end;
```

Memilih elemen dengan nilai paling rendah di bagian yang belum diurutkan

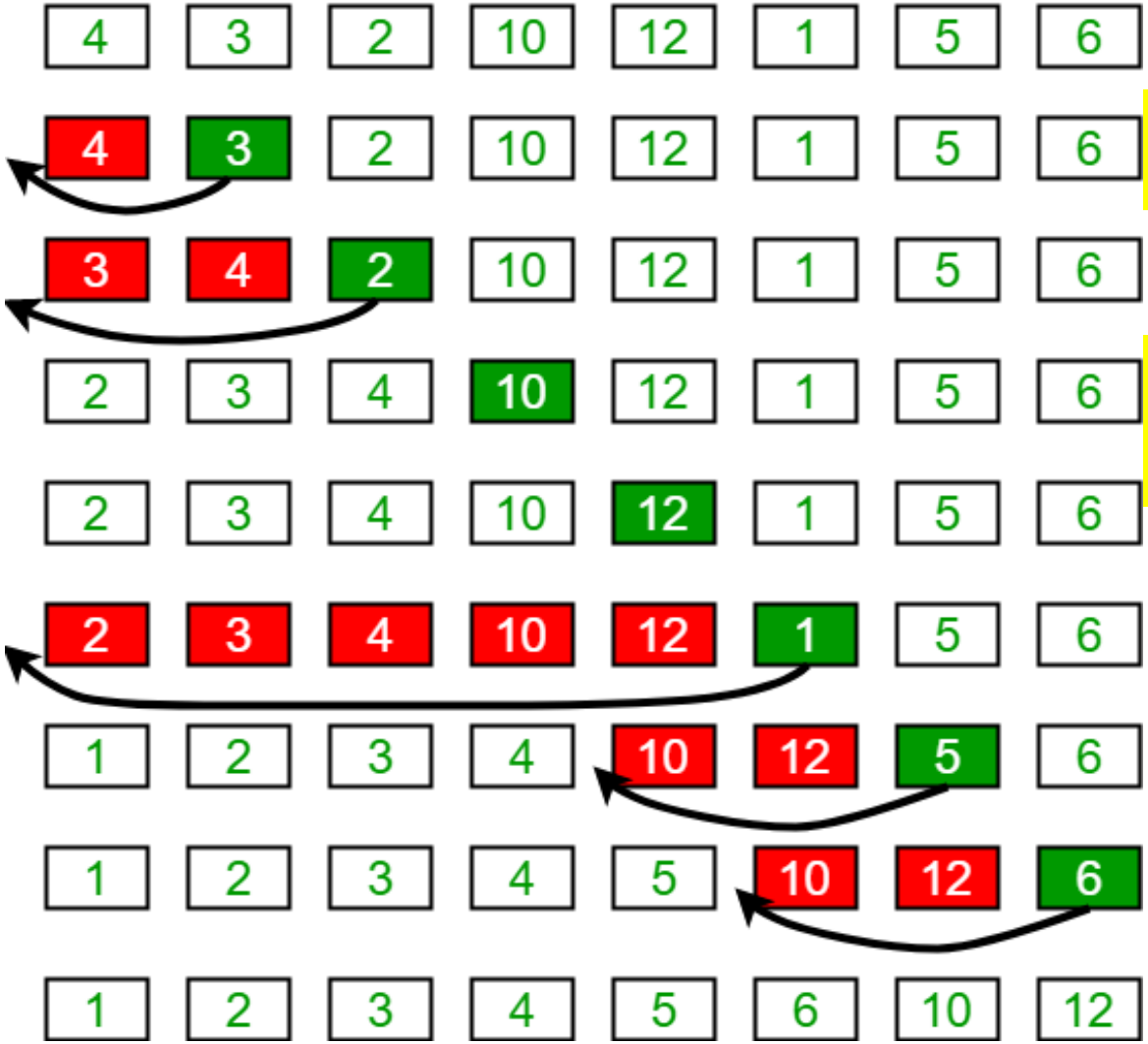
Menempatkan nilai terendah (Arr[idx_min]) di awal bagian yang belum diurutkan

INSERTION SORT

- Pengurutan yang dilakukan dengan cara menyisipkan elemen pada posisi yang sudah ditentukan atau yang seharusnya.
- Data diperiksa satu per satu mulai **dari yang kedua sampai dengan yang terakhir**. Apabila elemen yang diperiksa lebih kecil daripada elemen-elemen sebelumnya, maka **elemen tersebut disisipkan pada posisi yang sesuai**.

6 5 3 1 8 7 2 4

Data yang akan diurutkan:



Cek elemen ke-2 dan tempatkan pada posisi yang sesuai

Cek elemen ke-4, posisi sudah tepat jadi tidak perlu disisipkan di posisi lain

Data yang telah urut:

1 2 3 4 5 6 10 12

IMPLEMENTASI PROGRAM INSERTION SORT

```
Type Larik = array [1..100] of integer;
procedure Insertion_Sort(Arr:Larik; n:integer;);
var i,j,key: integer;
begin
    for i:= 2 to n do begin
        begin
            key:=Arr[i];
            j:=i-1;
            while(j>0 and Arr[j]>key) do
            begin
                Arr[j+1]=Arr[j];
                j:=j-1;
            end;

        Arr[j+1]=key;

    end;
end;
```

Periksa elemen satu per satu dari elemen ke i=2 sampai terakhir (n)

Terjadi pergeseran isi array apabila elemen yang diperiksa (key), lebih kecil daripada elemen-elemen sebelumnya

Tempatkan pada posisi yang seharusnya

Data yang akan diurutkan: 4 3 2 10 12 1 5 6

`i=2, key=3, j=1` → 4 3 2 10 12 1 5 6

`i=3, key=2, j=2,1` → 3 4 2 10 12 1 5 6

`i=4, key=10, j=3` → 2 3 4 10 12 1 5 6

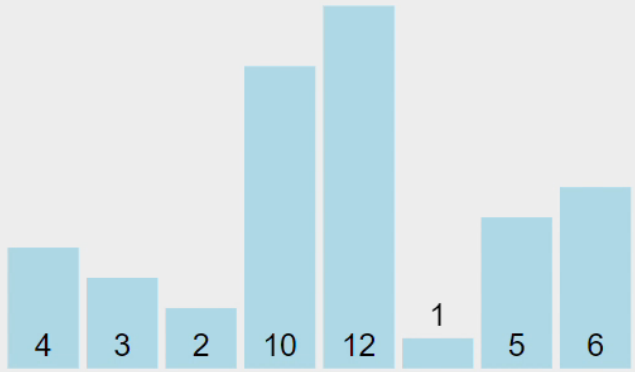
`i=5, key=12, j=4` → 2 3 4 10 12 1 5 6

`i=6, key=1, j=5..1` → 2 3 4 10 12 1 5 6

`i=7, key=5, j=6,5` → 1 2 3 4 10 12 5 6

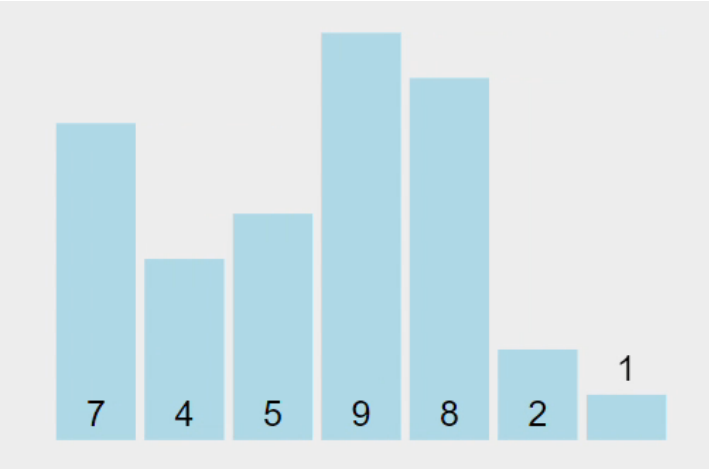
`i=8, key=6, j=7,6` → 1 2 3 4 5 10 12 6

Data yang telah urut: 1 2 3 4 5 6 10 12

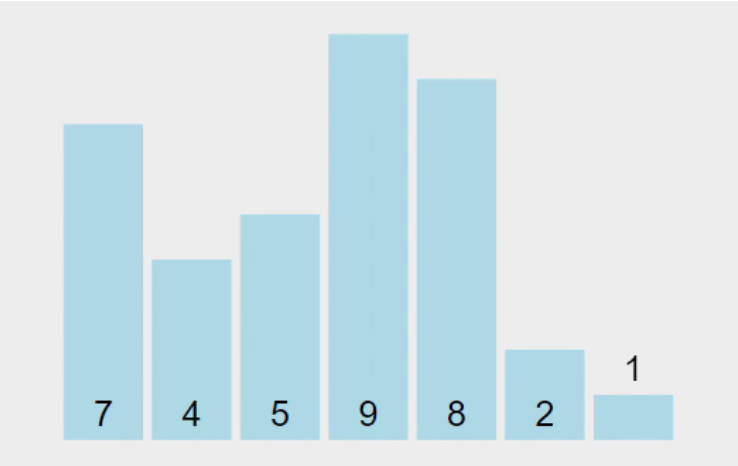


PERBANDINGAN ALGORITMA SORTING

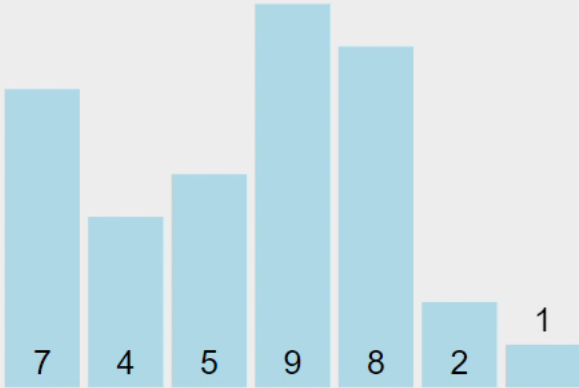
BUBBLE SORT



SELECTION SORT



INSERTION SORT



- Secara performa, ketiga prosedur pengurutan tersebut memiliki kompleksitas yang sama atau setara. Tidak ada yang lebih unggul atau lebih cepat dari yang lainnya.
- Tidak ada algoritma terbaik untuk setiap situasi, semuanya tergantung situasi dari setiap masalah. Banyak penelitian membandingkan beberapa algoritma sorting, hasilnya pun berbeda-beda.

POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH