

# ANALISIS ALGORITMA

Week 02: Kasus Algoritma Sorting Sederhana

PROGRAM PASCA SARJANA INFORMATIKA  
FAKULTAS TEKNIK INFORMATIKA  
UNIVERSITAS TELKOM

2022/2023

# Problem Sorting

atau Permutasi Data  
bagian 2 dari 4

Algoritma Insertion Sort

---

# Contoh: Insertion Sort



Diperkenalkan oleh John Mauchly(1946), Knuth(1973)

Idenya:

- Seperti bermain kartu
- Kartu baru disisipkan pada posisi yang sesuai,
- Sehingga kartu ditangan akan selalu terurut,
- Tidak bergantung pada nilai kartu di meja



43



81



17



74

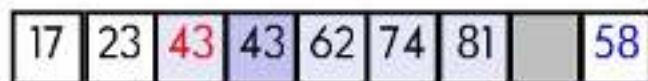
⋮



62



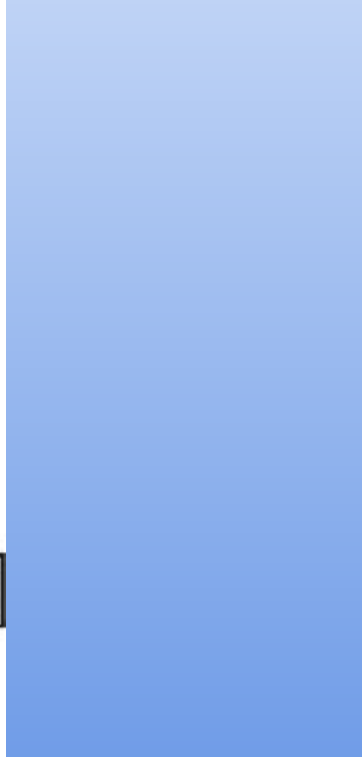
43



39

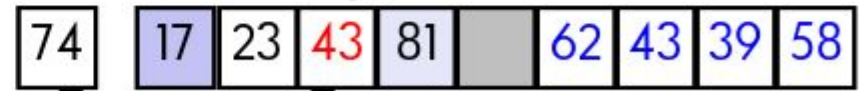
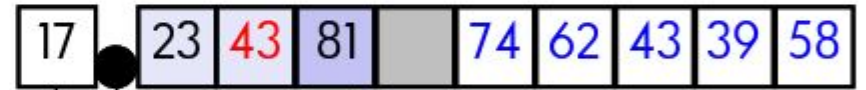


58





⋮



⋮

Selection sort vs. Insertion sort

# Karakteristik Insertion Sort

```
proc InsertionSort( A[1..n] )  
  i := 2  
  while i ≤ n do  
    t := A[i]  
    j := i-1  
    while j ≥ 1 and A[j] > t do  
      A[j+1] := A[j]  
      j := j - 1  
    endwhile  
    A[j+1] := t  
    i := i + 1  
  endwhile
```

- Seperti bermain kartu
- Kartu baru disisipkan pada posisi yang sesuai,
- Sehingga kartu ditangan akan selalu terurut,
- Tidak bergantung pada nilai kartu di meja



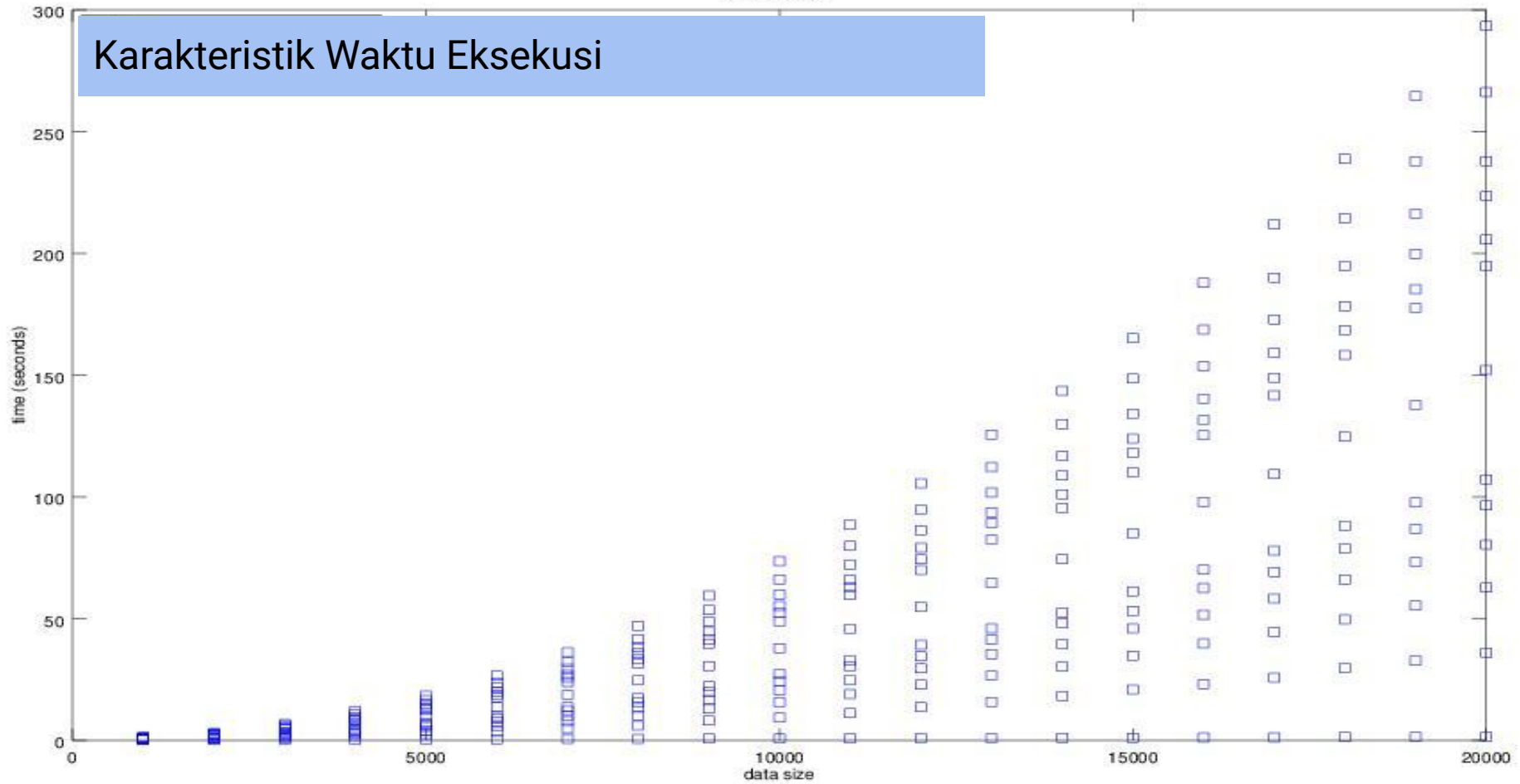
# Pembuktian Insertion Sort

```
proc InsertionSort( A[1..n] )
  i := 2
  { L.I.#1: . . . }
  while i ≤ n do
    t := A[i]
    j := i - 1
    { L.I.#2: . . . }
    while j ≥ 1 and A[j] > t do
      A[j+1] := A[j]
      j := j - 1
    endwhile
    { Term#2: . . . }
    A[j+1] := t
    i := i + 1
  endwhile
  { Term#1: . . . }
```

- Buktikan mulai dari loop yang lebih dalam baru kemudian loop luar
- Tentukan loop invarian dan kondisi terminasi yang tepat untuk masing-masing loop
- Buktikan dengan cara induksi
  - Basis
  - Induksi
  - Terminasi

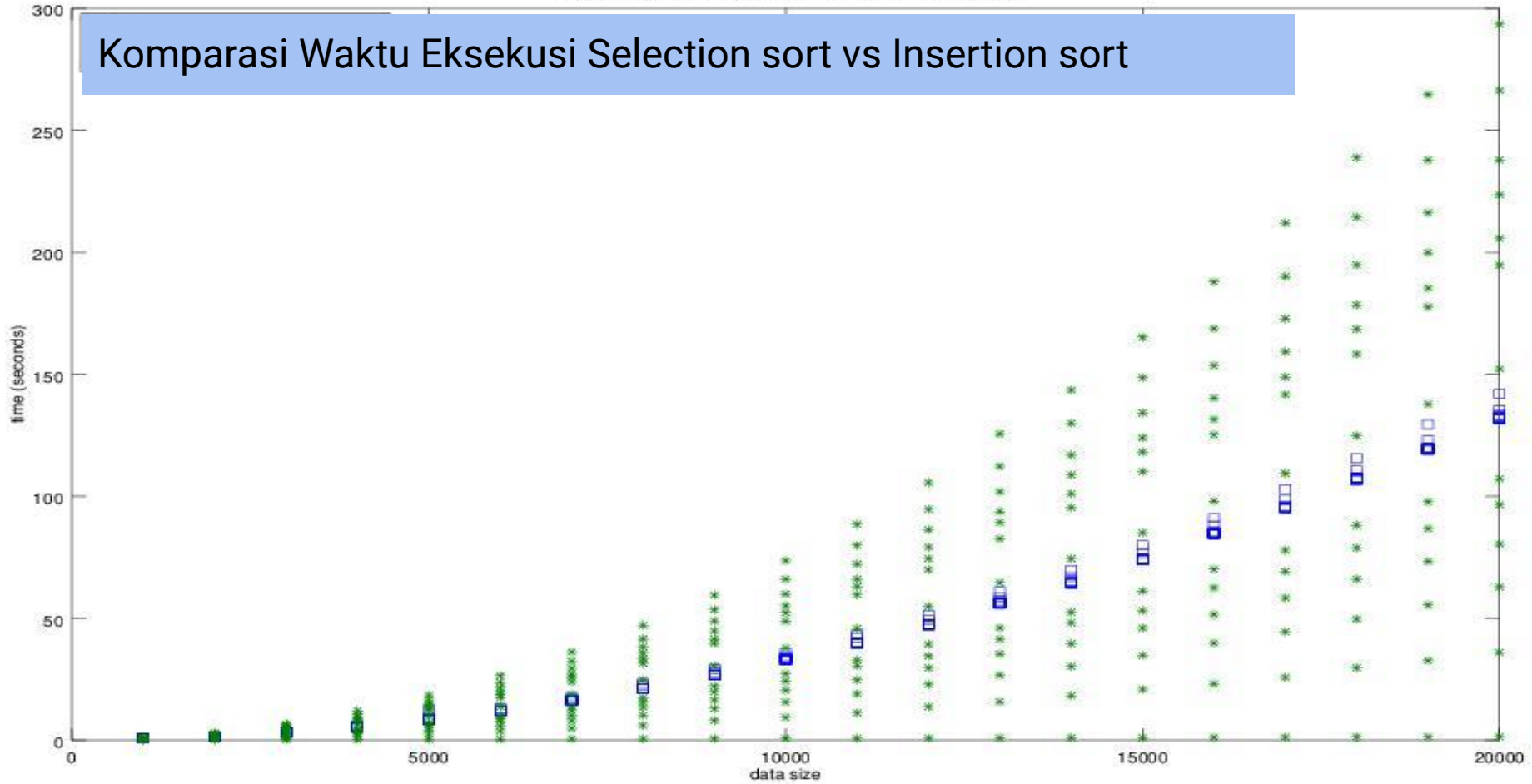


# Karakteristik Waktu Eksekusi



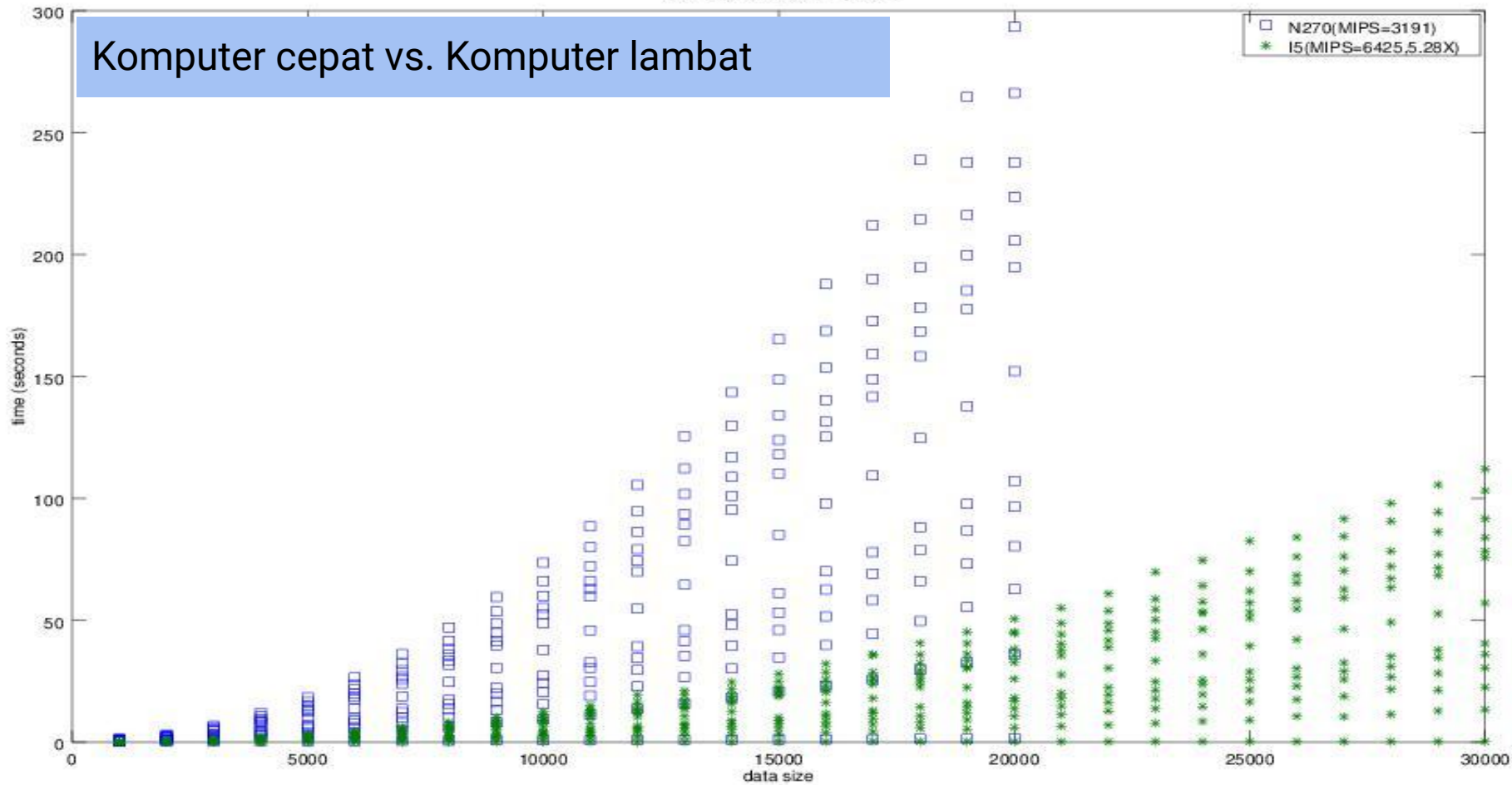


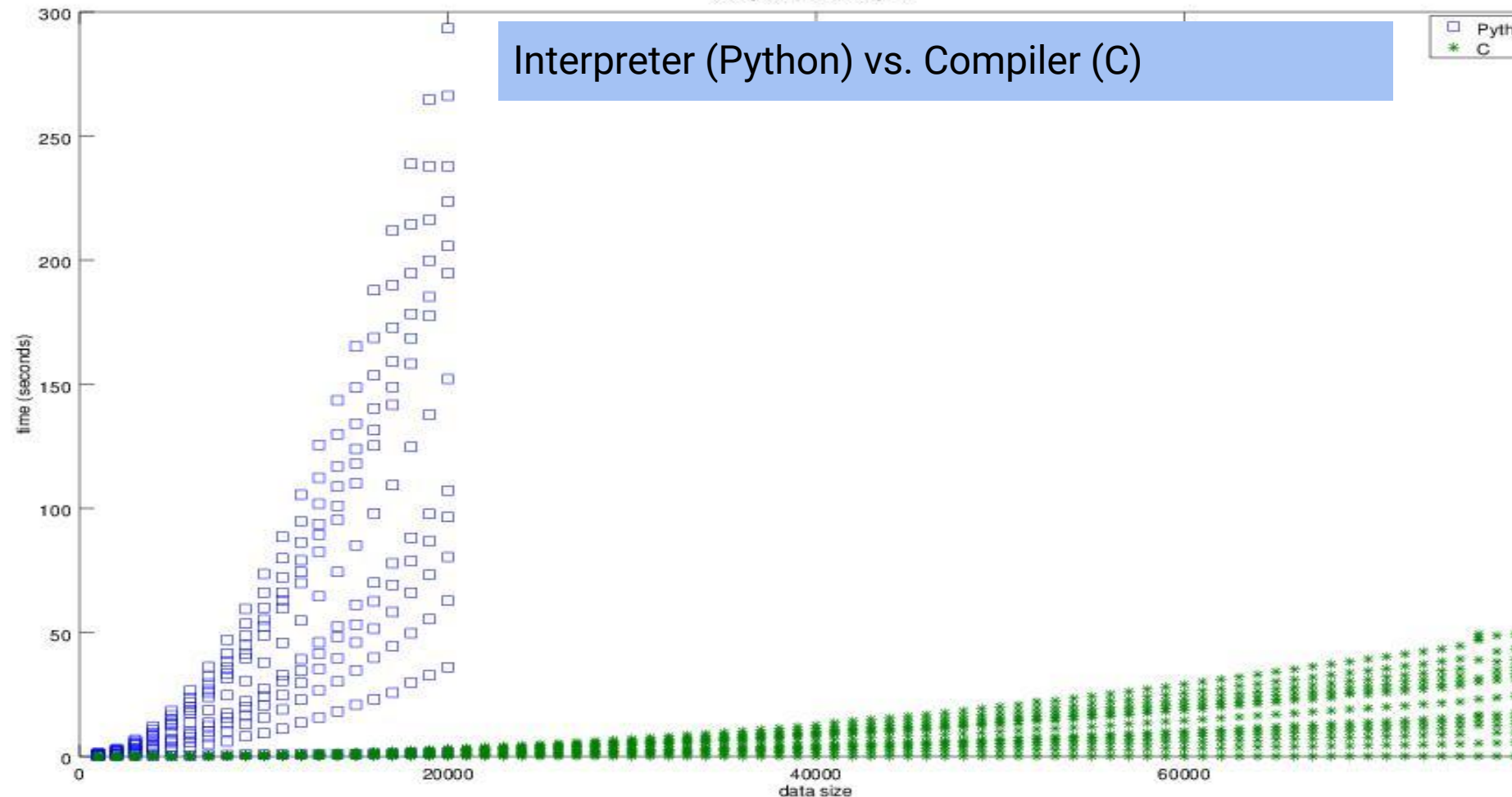
# Komparasi Waktu Eksekusi Selection sort vs Insertion sort



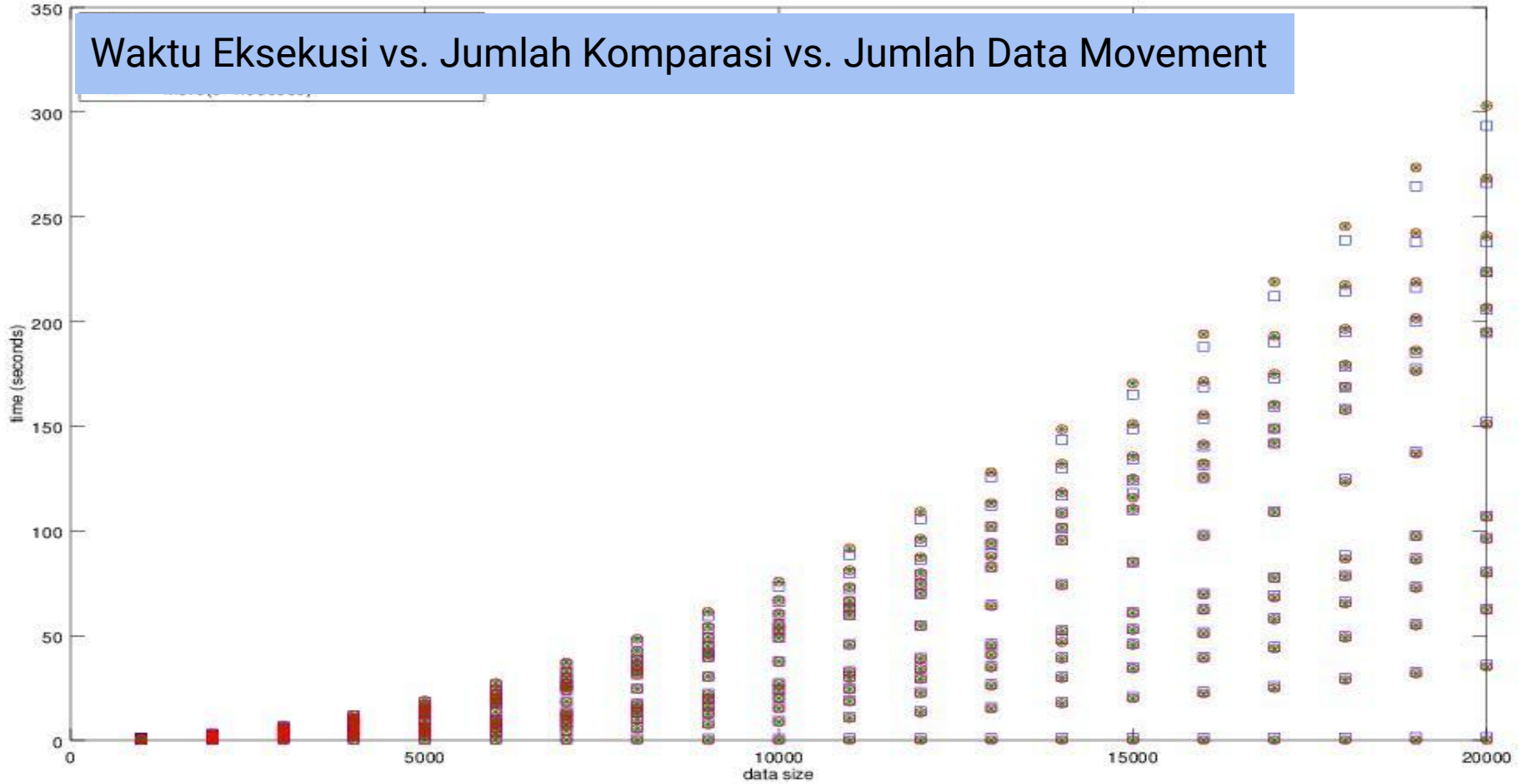
Slower vs. Faster Computers

Komputer cepat vs. Komputer lambat



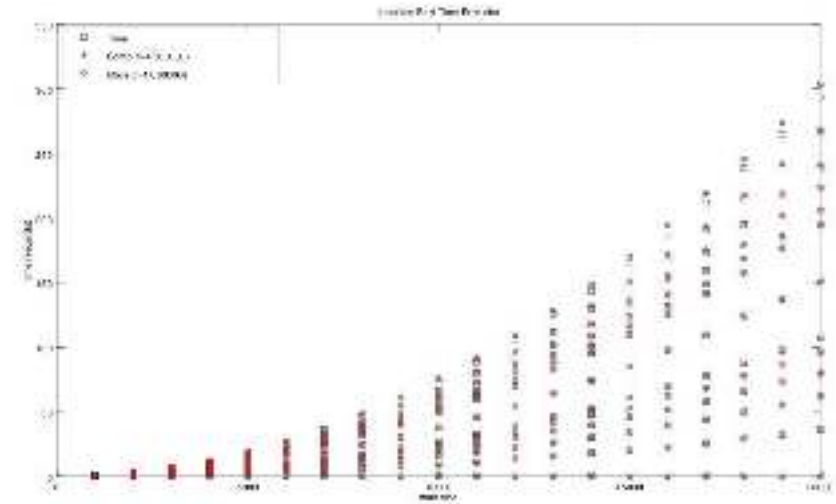


# Waktu Eksekusi vs. Jumlah Komparasi vs. Jumlah Data Movement

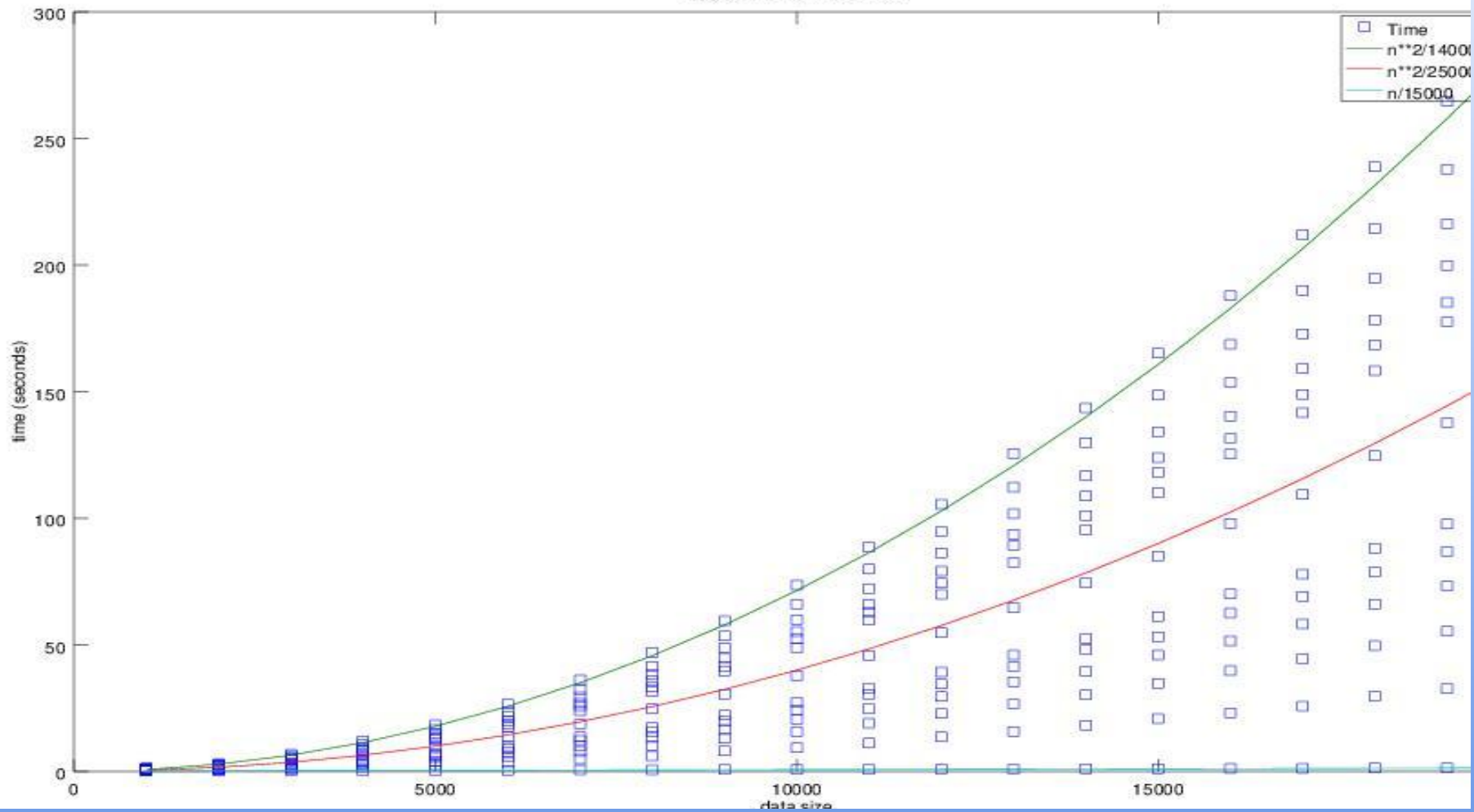


# Karakteristik Insertion Sort

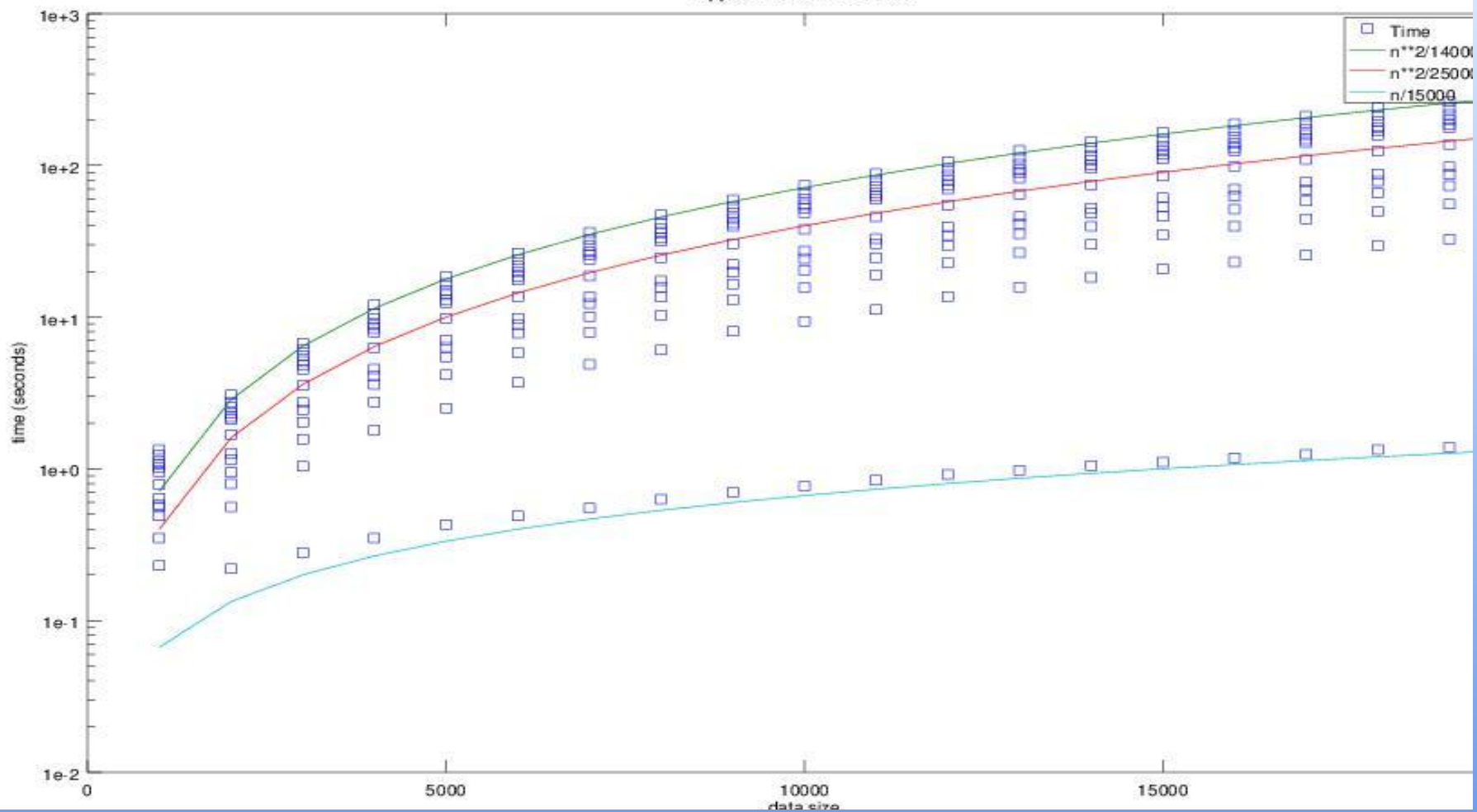
```
proc InsertionSort( A[1..n] )  
  i := 2  
  while i ≤ n do  
    t := A[i]  
    j := i-1  
    while j ≥ 1 and A[j] > t do  
      A[j+1] := A[j]  
      j := j - 1  
    endwhile  
    A[j+1] := t  
    i := i + 1  
  endwhile
```



Upper and Lower Bounded



Upper and Lower Bounded





# Utak Atik Hitungan Insertion Sort

```
proc InsertionSort( A[1..n] )
  i := 2
  while i ≤ n do
    t := A[i]
    j := i-1
    while j ≥ 1 and A[j] > t do
      A[j+1] := A[j]
      j := j - 1
    endwhile
    A[j+1] := t
    i := i + 1
  endwhile
```

- Jumlah operasi terdalam ( $A[j+1] := A[j]$ ) berada dalam 2 nested loops:
  - loop luar:  $i:=2 \dots n$ , atau  $n-1$  iterasi
  - loop dalam  $j:=i-1 \dots 1$  atau s.d.  $A[j] \leq t \leq A[j+2]$
- Worst case, loop dalam berhenti saat  $j == 0$ 
  - loop dalam  $i-1$  iterasi, sehingga  $T_{\text{ISORT}}(\mathbf{n}) = \sum_{i=2..n} (i-1) = \sum_{i=1..n-1} (i) = \frac{1}{2}(n-1+1)(n-1) = \frac{1}{2}(n)(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n$
- “Best” case, tidak masuk loop karena  $A[j==i-1] > t$ 
  - hanya loop luar, sehingga  $T_{\text{ISORT}}(\mathbf{n}) = n-1$
- Average case adalah rata<sup>2</sup> dari semua kemungkinan loop-dalam berhenti.
  - Asumsi probabilitas terjadinya loop-dalam antara 0 kali s.d.  $i-1$  kali adalah sama
  - Sehingga rerata jumlah iterasi dalam adalah  $T_j(i) = (\sum_{j=0..i-1} (j))/i = (\sum_{j=1..i} (j))/i - 1 = \frac{1}{2}(i+1) - 1 = \frac{1}{2}i - \frac{1}{2}$
  - $T_{\text{ISORT}}(\mathbf{n}) = \sum_{i=2..n} (\frac{1}{2}i - \frac{1}{2}) = \frac{1}{2} \sum_{i=2..n} (i-1) = \frac{1}{2}(\frac{1}{2}n^2 - \frac{1}{2}n) = \frac{1}{4}n^2 - \frac{1}{4}n$

# Kualitas Lain dari Insertion Sort

- Stabilitas
- Proses incremental
- Cepat untuk data “hampir terurut”

# Randomized Insertion Sort

```
proc InsertionSort( A[1..n] )  
  i := 2  
  while i ≤ n do  
    r := random(i, n)  
    t := A[r]  
    A[r] := A[i]  
    j := i-1  
    while j ≥ 1 and A[j] > t do  
      A[j+1] := A[j]  
      j := j - 1  
    endwhile  
    A[j+1] := t  
    i := i + 1  
  endwhile
```

- Mengurangi disparitas waktu karena ketergantungan data masukan
- Diskusikan + dan - nya!

# Akhir Bagian 2 dari Topik Minggu 02