

LAPORAN PRAKTIKUM
POSTTEST (1)
STRUKTUR DATA



Disusun oleh:
Khanza Humaira (2509106065)
Kelas (B1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2026

1. Kompleksitas Waktu Analisis Algoritma (SOAL-1)

Algoritma FindMin bertujuan mencari elemen dengan nilai terkecil dalam sebuah array berukuran n . Algoritma ini melakukan pencarian linear (linear search) karena memeriksa elemen satu per satu. Cara kerjanya adalah sebagai berikut:

1. Asumsikan elemen pertama sebagai nilai minimum sementara.
2. Bandingkan setiap elemen berikutnya dengan nilai minimum.
3. Jika ditemukan elemen yang lebih kecil, perbarui nilai minimum.
4. Setelah seluruh elemen diperiksa, nilai minimum dikembalikan.

2. Pseudocode

```
Procedure FindMin(A[0..n-1], n)
```

```
    min ← A[0]  
    indexMin ← 0
```

```
    for i ← 1 to n-1 do  
        if A[i] < min then  
            min ← A[i]  
            indexMin ← i  
        end if  
    end for
```

```
    return min, indexMin
```

```
End Procedure
```

Pseudocode	Cost	T_{min}	T_{max}
$\text{min} \leftarrow A[0]$	C1	1	1
$\text{for } i = 1 \text{ to } n-1$	C2	$n-1$	$n-1$
$\text{if } A[i] < \text{min}$	C3	$n-1$	$n-1$
$\text{min} \leftarrow A[i]$	C4	0	$n-1$
return min	C5	1	1
Total waktu (T_n)		$O(n)$	$O(n)$

Tabel 2.1: Tabel Pseudocode Analisis Kompleksitas

Penjelasan:

=> Best Case

$T_{min}(n) =$

$$C_1 + C_2(n-1) + C_3(n-1) + C_5$$

Gabungkan:

$T_{min}(n) =$

$$C_1 + C_5 + (C_2 + C_3)(n-1)$$

Misal:

$$a = C_1 + C_5$$

$$b = (C_2 + C_3)$$

Maka: $T_{min}(n) = a + b(n-1)$

Karena suku yang dominan adalah n,

$T_{min}(n) = O(n)$

=> Worst Case

$T_{max}(n) =$

$$C_1 + C_2(n-1) + C_3(n-1) + C_4(n-1) + C_5$$

Gabungkan:

$T_{max}(n) =$

$$C_1 + C_5 + (C_2 + C_3 + C_4)(n-1)$$

Misal:

$$a = C_1 + C_5$$

$$d = (C_2 + C_3 + C_4)$$

Maka:

$T_{max}(n) = a + d(n-1)$

Karena suku dominan tetap n,

$T_{max}(n) = O(n)$

3. Output

```
Nilai Minimum : 1
Indeks : 0
```

Gambar 3.1: Screenshot output nilai minimum dan indeks.