PEMBAHASAN UTS ASA

1) a. Kasus terbaik terjadi ketika $a_0 = 1$
Kasus terburuk terjadi ketika $a_n = 1$

b. $T_{min}(n) = 0$
$T_{max}(n) = n$

c. Kompleksitas linear → $O(n)$

2) a. Kompleksitas Konstan → $O(1)$
Berapapun ukuran data / masukan yang diterima, algoritma akan memiliki jumlah langkah yang sama untuk dieksekusi
Contoh :
Algoritma menambahkan elemen baru dalam linked list

```
→ void add_list (node *anchor, node *new_list)
{
    new_list -> next = achor -> next ;
    anchor -> next = new_list ;
}
```

b. Kompleksitas linear → $O(n)$
Algoritma dengan kompleksitas linear tumbuh selaras dengan pertumbuhan ukuran data.
Contoh :
Algoritma linear search pada python

```
→ def linear_search (list, search) :
      for i in range (0, len(list)) :
          if list [i] == search :
              print ("Nilai ditemukan pada posisi " + str(i))
              return 0
      print ("Nilai tidak ditemukan")
      return -1
```

c. Kompleksitas polinomial → $O(n^m)$
Algoritma tidak efisien, langkah penyelesaian jauh lebih besar daripada jumlah data.
Contoh :

```
def kali (a,b) :
    res = 0
    for i in range (a) :
```

for j in range (b) :
    res += 1
return res

3) a. $T(n) = 5n + 1024 = O(n)$

$5n + 1024 \leq 5n + 1024n$, $n \geq 1$

$\leq 1029n$

$T(n) \leq cf(n)$

$C = 1029$, $n_0 = 1$

b. $T(n) = 1 + 2 + 3 \dots + n = O(n^2)$

$n(n+1)/2 \leq O(n^2)$, $n \geq 1$

$(n^2 + n)/2 \leq O(n^2)$

$n^2/2 + n/2 \leq n^2 + n^2$

$\leq 2n^2$

$C_0 = 2$, $n = 1$

c. $T(n) = 4 \cdot 2^n + n^2 = O(n^2)$

$4 \cdot 2^n + n^2 \leq 4 \cdot n^2 + n^2$, $n \geq 1$

$(4 \cdot 2^n) + n^2 \leq 5(n^2)$

$C_0 = 5$, $n = 1$

d. $T(n) = \log n^5 = O(\log n)$

$5 \log n \leq 6 \log n$, $n \geq 1$

$C_0 = 6$ $n_0 = 1$

e. $T(n) = 8 \log 4^n = O(n)$

$n \times 8 \log 4 \leq 5n$, $n \geq 1$

$C = 5$, $n_0 = 1$

4) a. $T(n) = 2n^4 + 5n^3 + 10n + 4$

$2n^4 + 5n^3 + 10n + 4 \leq 2n^4 + 5n^4 + 10n^4 + 4n^4$

$\leq 21n^4$, $n \geq 1$

$C = 21$, $n_0 = 1$ Big O $= O(n^4)$

$2n^4 + 5n^3 + 10n + 4 \geq 2n^4$

$C = 2$ $n_0 = 1$

Omega $\Omega = (n^4)$

Jadi Big $\Theta = \Theta(n^4)$

b. $T(n) = 3n^2 + 2n \log n$

$3n^2 + 2n \log n \leq 3n^2 + 2n^2$

$\leq 5n^2$, $n \geq 1$

$C = 5$ $n_0 = 1$

Scanned by CamScanner

Big O = $O(n^2)$

$3n^2 + 2n\log n \geq 3n^2$

$C = 3$, $n_0 = 1$

Omega $\Omega = (n^2)$

Maka Big $\theta = \theta(n^2)$

c. $6n^3 + (\log n)^4 = O(n^3)$

$6n^3 + (\log n)^4 \leq Cn^3$

$\leq 6n^3 + n^3$

$\leq 7n^3$

$C_0 = 7$ dan $n_0 = 1$

Big O $= O(n^3)$

$6n^3 + (\log n)^4 \geq 6n^3$

$C = 6$, $n_0 = 1$

Omega $\Omega = (n^3)$ Mk Big $\theta = \theta(n^3)$

5) Relasi Rekurens :

$T(n) = 0$, $n = 0$

$T(n) = T(n-1) + 1$, $n > 0$

Kompleksitas Waktu

$T(n) = T(n-1) + 1$

$= [T(n-2) + 1] + 1 = T(n-2) + 2$ substitusikan $T(n-1) = T(n-2) + 1$

$= [T(n-3) + 2] + 1 = T(n-3) + 3$ substitusikan $T(n-2) = T(n-3) + 1$

$= \ldots$

$= T(0) + n$

$= 0 + n$

Jadi, $T(n) = n$

$T(n) \in O(n)$