LAPORAN PRAKTIKUM 3

ANALISIS ALGORITMA

DISUSUN OLEH

AHMAD IRFAN FADHOLI

140810180034

PROGRAM STUDI TEKNIK INFORMATIKA

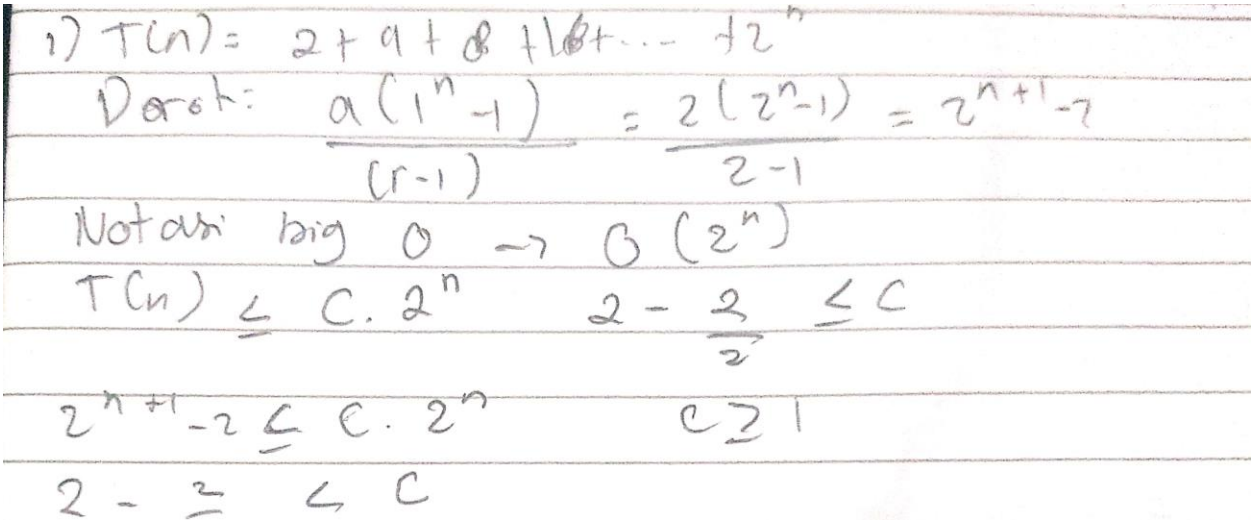
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

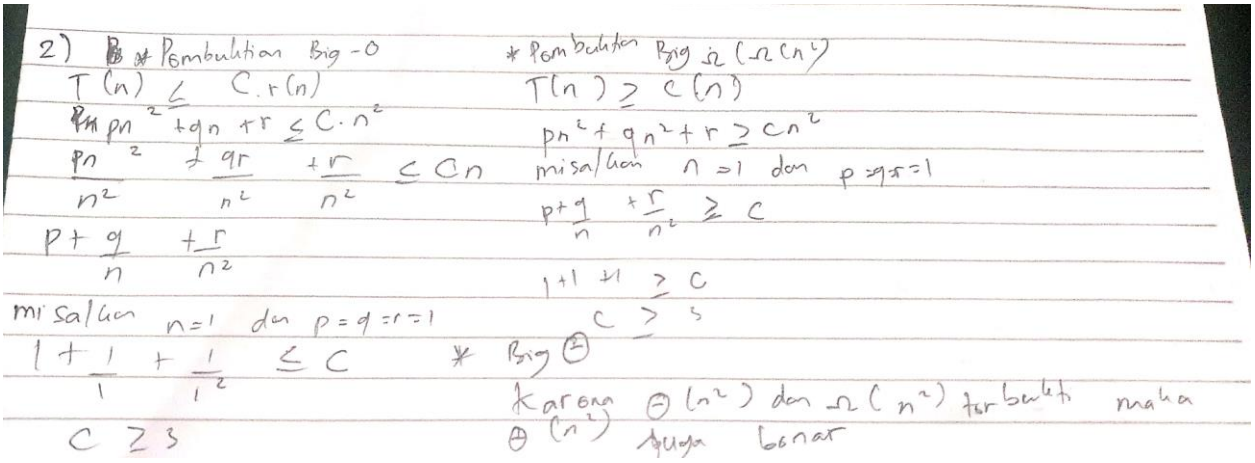
2020

Worksheet 3

1. Untuk $T(n) = 2 + 4 + 6 + 8 + 16 + \cdots + n^2$, tentukan nilai C, f(n), n_o, dan notasi Big-O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C$ *untuk semua* $n \geq n_0$



2. Buktikan bahwa untuk konstanta-konstanta positif p, q, dan r:
$T(n) = pn^2 + qn + r$ adalah $O(n^2), \Omega(n^2), dan\ \Theta(n^2)$



3. Tentukan waktu kompleksitas asimptotik (Big-O, Big-Ω, dan Big-Θ) dari kode program berikut:

```
for k ← 1 to n do
    for i ← 1 to n do
        for j ← to n do
            w_ij ← w_ij or w_ik and w_kj
        endfor
    endfor
endfor
```

3) $T(n) = O(n) + O(n) + O(n) + O(1)$
$= O(n^3) \rightarrow f(n)$
Big $O = O(f(n)) = O(n^3)$
Big $\Omega = \Omega(f(n)) = \Omega(n^3)$
Big $\Theta = \Theta(f(n)) = \Theta(n^3)$, karena Big O = Big Ω

4. Tulislah algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran n x n. Berapa kompleksitas waktunya T(n)? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big-Ω, dan Big-Θ?

4) For i ← 1 to n do
for j ← 1 to n do
hasil [i,j] ← a[i,j] + b[i,j]
endfor
endfor

$T(n) = O(n) + O(n) \cdot O(n^2) \rightarrow f(n)$
Big $O = O(n^2)$
Big $\Omega = \Omega(n^2)$
Big $\Theta = \Theta(n^2)$

5. Tulislah algoritma untuk menyalin (copy) isi sebuah larik ke larik lain. Ukuran elemen larik adalah n elemen. Berapa kompleksitas waktunya T(n)? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big-Ω, dan Big-Θ?

5) For i ← 1 to n do
$a_i \leftarrow b_i$
endfor

$T(n) = n$

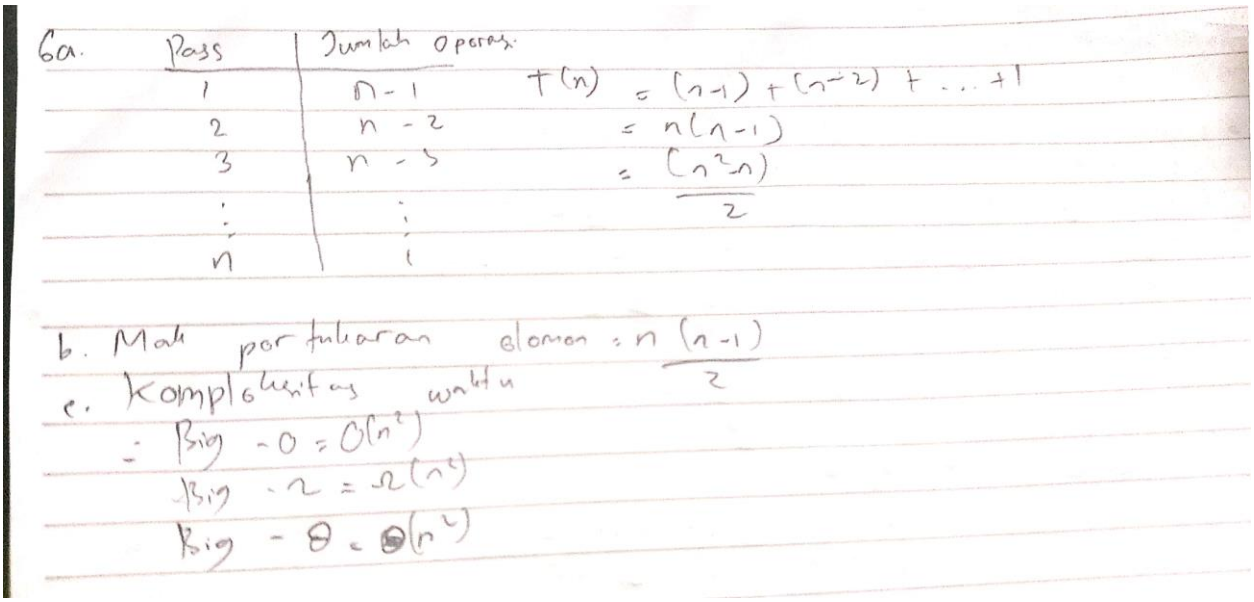
$O(n)$	$\Omega(n)$
$n \leq cn$	$n \geq cn$
$c \geq 1$	$c \leq 1$

$O(n) = \Omega(n) \rightarrow \Theta(n)$

6. Diberikan algoritma Bubble Sort sebagai berikut:

```
procedure BubbleSort(input/output a1, a2, ..., an : integer)
{ Mengurut tabel integer TabInt[1..n] dengan metode pengurutan bubble-
sort
  Masukan: a1, a2, ..., an
  Keluaran: a1, a2, ..., an (terurut menaik)
}
Deklarasi
   k : integer     { indeks untuk traversal tabel }
   pass : integer { tahapan pengurutan }
   temp : integer { peubah bantu untuk pertukaran elemen tabel }
Algoritma
   for pass ← 1 to n - 1 do
     for k ← n downto pass + 1 do
       if ak < ak-1 then
           { pertukarkan ak dengan ak-1 }
          temp ← ak
          ak ← ak-1
          ak-1←temp
       endif
     endfor
   endfor
```

(a) Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!
(b) Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?
(c) Hitung kompleksitas waktu asimptotik (Big-O, Big-Ω, dan Big-Θ) dari algoritma Bubble Sort tersebut!



7. Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:
   (a) Algoritma A mempunyai kompleksitas waktu O(log N)
   (b) Algoritma B mempunyai kompleksitas waktu O(N log N)
   (c) Algoritma C mempunyai kompleksitas waktu O(N^2)
   Untuk problem X dengan ukuran N=8, algoritma manakah yang paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?

7) a. Algoritma A → $O(\log n)$
b. Algoritma B → $O(n \log n)$
c. Algoritma C → $O(n^2)$

$N = 8$

A → $O(\log 8) = O(3 \log 2)$

B → $O(8 \log 8) = O(2^3 \log 2)$

C → $O(8^2) = O(64)$

A merupakan algoritma tercepat karena semakin kecil nilainya semakin dikit operasinya

8. Algoritma mengevaluasi polinom yang lebih baik dapat dibuat dengan metode Horner berikut:

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \ldots + x(a_{n-1} + a_n x)))\ldots))$$

```
function p2(input x : real) → real
{ Mengembalikan nilai p(x) dengan metode Horner}

Deklarasi
  k : integer
  b1, b2, ..., bn : real

Algoritma
  bn ← an
  for k ← n - 1 downto 0 do
     bk ← ak + bk+1 * x
  endfor
  return b0
```

Hitunglah berapa operasi perkalian dan penjumlahan yang dilakukan oleh algoritma diatas, Jumlahkan kedua hitungan tersebut, lalu tentukan kompleksitas waktu asimptotik (Big-O)nya. Manakah yang terbaik, algoritma p atau p2?

8) Algoritma P
Jumlah = n kali
Kali = n kali
$T(n) = n + n = 2n$
dari kodenya setara yaitu sama-sama bagus, namun P2 lebih baik karena kompleksitas waktunya lebih kecil

Algoritma P2
$T_2(n)$: [illegible]