



Lembar Kerja Responsi 4
Mata Kuliah KOM 401 Analisis Algoritme
Semester Genap Tahun Akademik 2020/2021

Asisten Praktikum: 1. M. Faishal Wicaksono R. 2. Zaki Muttaqin	Nama : Shibgotalloh Sabilana NIM : G64180002
---	---

A. Iteratif Sederhana

Berikut ini terdapat potongan program yang memuat struktur perulangan iteratif sederhana. Tentukan kompleksitas big-Oh dari setiap potongan program tersebut.

No	Potongan Program	Big-Oh
1	<pre>int n; scanf("%d", &n); for(int i = n; i > 0; i=i/2){ // Suatu pernyataan konstan O(1) break; }</pre>	$O(1)$
2	<pre>for(int i = 1; i < n; i=i*2){ // Suatu pernyataan konstan O(1) }</pre>	$O(\log_2 n)$
3	<pre>int n; scanf("%d", &n); while (n > 0) { n = n / 2; }</pre>	$O(\log_2 n)$

4	<pre>int n; scanf("%d", &n); for(i = 0; i < sqrt(n); i++){ // Suatu pernyataan konstan O(1) }</pre>	$O(\sqrt{n})$
---	--	---------------

5	<pre>for(int i = n; i > 1; i=i--){ printf("%d", &i); }</pre>	$O(n)$
6	<pre>for(int i = 2; i < n; i=pow(i, 2)){ // Suatu pernyataan konstan O(1) }</pre>	$O(\log(\log_2 n))$
7	<pre>int n; scanf("%d", &n); for(i = 1; i < sqrt(n); i = i*2){ // Suatu pernyataan konstan O(1) }</pre>	$O(\log_2 \sqrt{n})$
8	<pre>int n; scanf("%d", &n); for(i = 2; i < n; i = i*2){ // Suatu pernyataan konstan O(n) }</pre>	$O(n \log_2 n)$
9	<pre>int n; scanf("%d", &n); for(i = 0; i < n+n+n; i = i++){ // Suatu pernyataan konstan O(n^2) }</pre>	$O(n^3)$
10	<pre>int n; scanf("%d", &n); for(i = 0; i < n*n; i = i++){ // Suatu pernyataan konstan O(1) }</pre>	$O(n^2)$

B. Iteratif Majemuk

Berikut ini terdapat potongan program yang memuat struktur perulangan iteratif majemuk. Tentukan kompleksitas big-Oh dari setiap potongan program tersebut.

No	Potongan Program	Big-Oh
1	<pre> for (int i=0; i<n; i++){ for (int j=0; j<n; j++){ printf("*"); } } </pre>	$O(n^2)$
2	<pre> for (int i=0; i<n; i++){ for (int j=0; j<sqrt(n); j++){ // Suatu pernyataan konstan O(1) } } </pre>	$O(n \cdot \sqrt{n})$
3	<pre> int n; int count = 0; scanf("%d", &n); for(int i = 0; i < n; i++){ while(c < i){ sum = sum + count; count++; } } </pre>	$O(n)$
4	<pre> int n; int c = 0; scanf("%d", &n); for(int i = 0; i < n; i++){ c = 0; while(c < i){ sum = sum + c; c++; } } </pre>	$O(n)$
5	<pre> scanf("%d", &n); for(int i = 0; i<n=pow(n,3); i++){ if(i<n=pow(n,3)){ printf("%d", i); } } </pre>	$O(n^3)$

6	<pre> int n; int count = 0; scanf("%d", &n); for(int i = 0; i < n; i++){ for(int j = 0; j < n*n; j++){ for(int k = n; k > 0; k--){ count = count + k; } } } </pre>	$O(n^4)$
7	<pre> int count = 0; for (int i=n/2; i<=n; i++){ for (int j=1; j+n/2<=n; j = j++){ for (int k=1; k<=n; k = k * 2){ count++; } } } </pre>	$O(n^2 \log n)$
8	<pre> for (int i=0; i<=sqrt(n); i++){ for (int j=n; j>=1; j = j--){ for (int k=1; k<=n; k = k * 2){ printf("*"); } } } </pre>	$O(n \sqrt{n} \log_2 n)$
9	<p>Kadane's Algorithm</p> <pre> int arr[M] = { //M element } //array dengan M element int top = arr[0]; int best = arr[0]; for(int i = 0; i < M; i++){ top = max(arr[i],top+arr[i]); best = max(top,best); } printf("%d", best); </pre>	$O(M)$
10	<pre> scanf("%d", &n); for(int i = 0; i < n*n; i++){ for(int j = m/4; j < m/2; i*=2) printf("I"); for(int j = 0; j < 2*m; i+=2){ printf("B"); for(int k = 0; j < 2*m; i+=2) printf("P"); } } for(int i = 0; i/2 < n*m; i*=2) printf("B"); </pre>	$O(m^2 * n^2)$

