



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

**Asisten Praktikum:**

- 1. Alfian Hamam Akbar**
- 2. Hilmi Farhan Ramadhani**

---

**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", &amp;n); for(int i = n; i &gt; 0; i=i/2){     // Suatu pernyataan konstan O(1)     break; }</pre>	$O(1)$
2	<pre>for(int i = 1; i &lt; n; i=i*2){     // Suatu pernyataan konstan O(1) }</pre>	$O(\log(n))$
3	<pre>int n; scanf("%d", &amp;n); while (n &gt; 0) {     n = n / 2; }</pre>	$O(\log(n))$
4	<pre>int n; scanf("%d", &amp;n); for(i = 0; i &lt; sqrt(n); i++){     // Suatu pernyataan konstan O(1) }</pre>	$O(\sqrt{n})$

5	<pre>for(int i = n; i &gt; 1; i=i--){     printf("%d", &amp;i); }</pre>	$O(n)$
6	<pre>for(int i = 2; i &lt; n; i=pow(i, 2)){     // Suatu pernyataan konstan O(1) }</pre>	$O(\log(\log(n)))$
7	<pre>int n; scanf("%d", &amp;n); for(i = 1; i &lt; sqrt(n); i = i*2){     // Suatu pernyataan konstan O(1) }</pre>	$O(\log(\sqrt{n}))$
6	<pre>int n; scanf("%d", &amp;n); for(i = 2; i &lt; n; i = i*2){     // Suatu pernyataan konstan O(n) }</pre>	$O(n \log n)$
9	<pre>int n; scanf("%d", &amp;n); for(i = 0; i &lt; n+n+n; i = i++){     // Suatu pernyataan konstan O(n^2) }</pre>	$O(n^3)$
10	<pre>int n; scanf("%d", &amp;n); for(i = 0; i &lt; 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&lt;n; i++){     for (int j=0; j&lt;n; j++){         printf("*");     } }</pre>	$O(n^2)$
2	<pre>for (int i=0; i&lt;n; i++){     for (int j=0; j&lt;sqrt(n); j++){         // Suatu pernyataan konstan O(1)     } }</pre>	$O(n\sqrt{n})$
3	<pre>int n; int count = 0; scanf("%d", &amp;n); for(int i = 0; i &lt; n; i++){     while(c &lt; i){         sum = sum + count;         count++;     } }</pre>	$O(n)$
4	<pre>int n; int c = 0; scanf("%d", &amp;n); for(int i = 0; i &lt; n; i++){     c = 0;     while(c &lt; i){         sum = sum + c;         c++;     } }</pre>	$O(n^2)$
5	<pre>scanf("%d", &amp;n); for(int i = 0; i&lt;n=pow(n,3); i++){     if(i&lt;n=pow(n,3)){         printf("%d", i);     } }</pre>	$O(n^3)$

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