

**LAPORAN PRAKTIKUM
ANALISIS ALGORITMA**



Disusun Oleh:

Firmansyah Yanuar 140810170051

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2018/2019**

1. A.

```

1  for j=1 to n-1
2      k = j                                // n-1 kali
3      for i=j+1 to n
4          if a[i] < a[k] then                // 1+2+3+...+(n-1) kali
5              k=i                            // 1+2+3+...+(n-1) kali
6          endif
7      endfor
8      tm = a[j]                             // n-1 kali
9      a[j] = a[k]                           // n-1 kali
10     a[k] = tm                             // n-1 kali
11 endfor

```

$$1 + 2 + 3 + \dots + (n-1) = \frac{n}{2}(n-1)$$

$$T(n) = (n-1) + \frac{n}{2}(n-1) + \frac{n}{2}(n-1) + (n-1) + (n-1) + (n-1)$$

$$T(n) = (4n-4) + n^2 - n$$

$$T(n) = n^2 + 3n - 4$$

$$T(n) \leq c.f(n)$$

$$n^2 + 3n - 4 \leq c.n^2$$

$$1 + \frac{3}{n} - \frac{4}{n^2} \leq c, \text{ misal } n_0 = 1$$

$$1 + \frac{3}{1} - \frac{4}{1} \leq c$$

$$c \geq 0$$

Maka kompleksitas O dari algoritma tersebut adalah $O(n^2)$ dengan $n_0 = 1$ dan $c \geq 0$

B.

```

1  for i=0 to n-1
2      for j=0 to n-1
3          c[i,j]=0                        //n*n kali
4          for k=0 to n-1
5              cij = d[i,k] and b[k,j]      //n*n*n kali
6              c[i,j] = c[i,j] or cij        //n*n*n kali
7          endfor
8      endfor
9  endfor

```

$$T(n) = n^2 + n^3 + n^3$$

$$T(n) = 2n^3 + n^2$$

$$T(n) \leq c.f(n)$$

$$2n^3 + n^2 \leq c.n^3$$

$$2 + \frac{1}{n} \leq c, \text{ misal } n_0 = 1$$

$$c \geq 3$$

Maka kompleksitas O dari algoritma tersebut adalah $O(n^3)$ dengan $n_0 = 1$ dan $c \geq 3$

2. A. Algoritma A

```

1  ada = 0;           // 1 kali
2  kx = 1;            // 1 kali
3  input br;          // 1 kali
4  for (i = 1; i < n+1; i++){
5      if (a[i] == br && (!ada)){ // n kali
6          ada = 1;           // 1 kali
7          kx = i;             // 1 kali
8          i = n+1;           // 1 kali
9      }
10 }
```

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

$$T(n) = n + 6$$

$$T(n) \leq c \cdot f(n)$$

$$n + 6 \leq c \cdot n$$

$$1 + \frac{6}{n} \leq c, \text{ misal } n_0 = 1$$

$$c \geq 7$$

Maka kompleksitas O dari algoritma tersebut adalah $O(n)$ dengan $n_0 = 1$ dan $c \geq 7$

A. Algoritma B

```

1  L = 1;             // 1 kali
2  R = n;             // 1 kali
3  ada = 0;           // 1 kali
4  input br;          // 1 kali
5  while((L <= R) && (!ada)){
6      m = (L+R) div 2; // log n kali
7      if(a[m] == br)   // log n kali
8          ada = 1;     // 1 kali
9      else if (br < a[m]) // log n kali
10         R = m-1;      // log n - 1 kali
11     else
12         L = m+1;
13 }
```

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

$$T(n) = 4 \log n + 4$$

$$T(n) \leq c \cdot f(n)$$

$$4 \log n + 4 \leq c \cdot \log n$$

$$4 + \frac{4}{\log n} \leq c, \text{ misal } n_0 = 10$$

$$4 + \frac{4}{\log 10} \leq c$$

$$4 + 4 \leq c$$

$$c \geq 8$$

Maka kompleksitas O dari algoritma tersebut adalah $O(\log n)$ dengan $n_0 = 10$ dan $c \geq 8$

B. Jumlah Instruksi Algoritma A = $O(10^8)$

Jumlah Instruksi Algoritma B = $O(\log 10^8) = O(8)$

Running time A = $\frac{10^8}{10^9} = 0.1 \text{ detik}$

Running time B = $\frac{8}{10^7} = 8 \times 10^{-7} \text{ detik}$

Maka algoritma B lebih baik daripada algoritma A karena running time B lebih cepat.