Worksheet 2
Analisis Algoritma

Rio Sapta Samudera
140810180030

Program Studi Teknik Informatika
Fakultas Matematika dan Ilmu Pengetahuan Alam
UNIVERSITAS PADJADJARAN

Worksheet 2

Studi Kasus 1: Pencarian Nilai Maksimal
Buatlah programnya dan hitunglah kompleksitas waktu dari algoritma berikut:
Algoritma Pencarian Nilai Maksimal

```
procedure CariMaks(input x1, x2, ... xn: integer, output maks:
integer)
{ Mencari elemen terbesar dari sekumpulan elemen larik integer x1, x2,
..., xn. Elemen terbesar akan
disimpan di dalam maks
Input: x1, x2, ..., xn
Output: maks (nilai terbesar)
}

Deklarasi
i : integer
Algoritma
maks <- x1
i <- 2
while i ≤ n do
     if xi > maks then
     Maks <-  xi
endif
i <- i + 1
endwhile
```

Jawaban Studi Kasus 1

A. Operasi Assignment
   maks <- x1 1 kali
   i <- 2 1 kali
   maks <- xi n kali
   i <- i + 1 n kali

   t1 = 1 + 1 + n + n = 2+2n (ii)
B. Operasi Perbandingan
   if xi > maks then n kali

t2 = n

C. Operasi Penjumlahan
i + 1 n kali

t3 = n

Jadi, T(n) = t1 + t2 + t3 = 2 + 2n + n + n = 2 + 4n

Studi Kasus 2: Sequential Search

Diberikan larik bilangan bulat x1, x2, … xn yang telah terurut menaik dan tidak ada elemen ganda. Buatlah programnya dengan C++ dan hitunglah kompleksitas waktu terbaik, terburuk, dan rata-rata dari algoritma pencarian beruntun (sequential search). Algoritma sequential search berikut menghasilkan indeks elemen yang bernilai sama dengan y. Jika y tidak ditemukan, indeks 0 akan dihasilkan.

```
procedure SequentialSearch(input x1, x2, … xn : integer, y : integer, output idx : integer)
{ Mencari y di dalam elemen x1, x2, … xn. Lokasi (indeks elemen) tempat ditemukan diisi ke dalam idx. Jika tidak ditemukan, makai idx diisi dengan 0.
Input: x1, x2, … xn
Output: idx
}

Deklarasi
i : integer
found : boolean {bernilai true jika y ditemukan atau false jika y tidak ditemukan}
Algoritma
i <- 1
found <- false
while (i ≤ n) and (not found) do
      if xi = y then
            found <- true
      else
            i <- i + 1
      endif
endwhile
{i < n or found}
```

```
If found then {y ditemukan}
     idx  i
else
     idx  0 {y tidak ditemukan}
endif
```

Jawaban Studi Kasus 2

A. Terbaik (Tmin(n))
   Operasi Assignment = 4
   Operasi Perbandingan = 2

   Tmin(n) = 4 + 2 = 6 = Ω(n)

B. Terburuk (Tmax(n))
   Operasi Assignment = 3 + n
   Operasi Perbandingan = 1 + n
   Operasi Penambahan = n

   Tmax(n) = 3 + n + 1 + n + n = 4 + 3n = O(n)

C. Rata-rata (Tavg(n))
   Tavg(n) = (T(min) + T(max))/2 = (6 + 4 + 3n)/2 = (10 + 3n)/2 = Θ(n)

Studi Kasus 3: Binary Search

Diberikan larik bilangan bulat x1, x2, … xn yang telah terurut menaik dan tidak ada elemen ganda. Buatlah programnya dengan C++ dan hitunglah kompleksitas waktu terbaik, terburuk, dan rata-rata dari algoritma pencarian bagi dua (binary search). Algoritma binary search berikut menghasilkan indeks elemen yang bernilai sama dengan y. Jika y tidak ditemukan, indeks 0 akan dihasilkan.

```
procedure BinarySearch(input x1, x2, ... xn : integer, x : integer,
output : idx : integer)

{ Mencari y di dalam elemen x1, x2, ... xn. Lokasi (indeks elemen)
tempat y ditemukan diisi ke dalam idx. Jika y tidak ditemukan makai dx
diisi dengan 0.
Input: x1, x2, ... xn
Output: idx
}
```

```
Deklarasi
i, j, mid : integer
found : Boolean
Algoritma
i <- 1
j <- n
found <- false
while (not found) and ( i ≤ j) do
      mid <- (i + j) div 2
      if xmid = y then
            found  true
      else
            if xmid < y then      {mencari di bagian kanan}
                  i <- mid + 1
            else                  {mencari di bagian kiri}
                  j <- mid – 1
            endif
      endif
endwhile

{found or i > j }
If found then
      Idx  mid
else
      Idx  0
endif
```

Jawaban Studi Kasus 3

A. Terbaik (Tmin(n))
   Operasi Assignment = 6
   Operasi Perbandingan = 2

   Tmin(n) = 6 + 2 = 8 = Ω(n)

B. Terburuk (Tmax(n))
   Karena ini adalah binary search, maka:
   Iterasi 1 = n
   Iterasi 2 = n/2
   Iterasi 3 = $n/2^2$
   Iterasi x = $n/2^x$

Hingga array menjadi 1

Maka:
n/2^k = 1
n = 2^k
log2(n) = log2(2^k)
log2(n) = k log2(2)
k = log2(n)

sehingga:
Tmax(n) = O(log2(n))

C. Rata-rata (Tavg(n))
Tavg(n) = (T(min) + T(max))/2 = (8 + log2(n))/2 = Θ(n)

Studi Kasus 4: Insertion Sort

1. Buatlah program insertion sort dengan menggunakan bahasa C++
2. Hitunglah operasi perbandingan elemen larik dan operasi pertukaran pada algoritma insertion sort.
3. Tentukan kompleksitas waktu terbaik, terburuk, dan rata-rata untuk algoritma insertion sort.

```
procedure InsertionSort(input/output x1, x2, ... xn : integer)
{ Mengurutkan elemen-elemen x1, x2, ...  xn dengan metode insertion sort.
Input: x1, x2, ... xn
Output: x1, x2, ... xn(sudah terurut menaik)
}

Deklarasi
i, j, insert : integer
Algoritma
for i <- 2 to n do
     insert <- xi
     j <- i
     while (j < i) and (x[j-i] > insert) do
          x[j] <- x[j-1]
          j <- j-1
     Endwhile
```

```
      x[j] = insert
endfor
```

Jawaban Studi Kasus 4

Jumlah Operasi:

1. Operasi Perbandingan = 2*((n-1) + (n-1)) = 2*(2n-2) = 4n - 4
2. Operasi Pertukaran = (n-1) * n = (n^2)-n

A. Terbaik (Tmin(n))
   Tmin(n) = 4n - 4 + 1 = 4n - 3 = Ω(n)

B. Terburuk (Tmax(n))
   Tmax(n) = 4n - 4 + (n^2) - n = (n^2) + 3n - 4 = O(n^2)

C. Rata-rata (Tavg(n))
   Tavg(n) = (Tmin(n) + Tmax(n)) / 2 = (n + n^2) / 2 = **Θ**(n^2)

Studi Kasus 5: Selection Sort

1. Buatlah program selection sort dengan menggunakan bahasa C++
2. Hitunglah operasi perbandingan elemen larik dan operasi pertukaran pada algoritma selection sort.
3. Tentukan kompleksitas waktu terbaik, terburuk, dan rata-rata untuk algoritma insertion sort.

```
Procedure SelectionSort(input/output x1, x2, ... xn : integer)
{ Mengurutkan elemen-elemen x1, x2, ...  xn dengan metode insertion sort.
Input: x1, x2, ... xn
Output: x1, x2, ... xn(sudah terurut menaik)
}

Deklarasi
i, j, imaks, temp : integer
Algoritma
for i <- n down to 2 do {pass sebanyak n-1 kali}
      imaks <- 1
      for j <- 2 to i do
            if xj > ximaks then
                  imaks <- j
            endif
      endfor
{pertukarkan ximaks dengan xi}
```

```
temp <- xi
xi <- ximaks
ximaks <- temp
```

Jawaban Studi Kasus 5

Jumlah Operasi:

1. Operasi Perbandingan =

$$\sum_{i=1}^{n-1} i = \frac{(n-1)+1}{2}(n-1) = \frac{1}{2}n(n-1) = \frac{1}{2}(n^2 - n)$$

2. Operasi Pertukaran = (n-1)

D. Terbaik (Tmin(n))
   Tmin(n) = ½ (n^2 - n) + 1 = Ω(n^2)

E. Terburuk (Tmax(n))
   Tmax(n) = ½ (n^2 - n) + (n-1) = O(n^2)

F. Rata-rata (Tavg(n))
   Tavg(n) = (Tmin(n) + Tmax(n)) / 2 = (n^2 + n^2) / 2 = **Θ**(n^2)

Program C++

Studi Kasus 1

```cpp
/*

Nama    = Rio Sapta Samudera

NPM     = 140810180030

Kelas   = B

*/

#include <iostream>

using namespace std;
```

```
int main()

{

    int i=0;

    int n=10;

    int x[n]{1,20,12,41,10,32,41,22,92,70};

    int maks = x[0];



    do{

        if(x[i]>maks){

            maks=x[i];

        }

        i++;

    }while(i<n);



    cout<<"Maks : "<<maks;



    return 0;

}
```

Studi Kasus 2

```
/*

Nama    = Rio Sapta Samudera
```

```
NPM     = 140810180030

Kelas   = B

*/

#include <iostream>

using namespace std;



int main()

{

    int i = 0;

    bool found = false;

    int n=10;

    int x[n]{10,9,8,7,6,5,4,3,2,1};

    int y;

    cout<<"enter number : ";

    cin>>y;



    do{

        if(x[i]==y){

            found=true;

            cout<<"founded in array number "<<i;

        }

        else
```

```
            i++;

    }while((i<n) && (not found));



    return 0;

}
```

Studi Kasus 3

```
/*

Nama    = Rio Sapta Samudera

NPM     = 140810180030

Kelas   = B

*/

#include <iostream>

using namespace std;



int main(){

    int i,j,y,mid;

    bool found;



    int n=10;

    int x[n]{1,19,27,36,45,54,63,72,81,90};

    i=0;
```

```
j=n;
found=false;

cout<<"enter number : ";
cin>>y;

do{
    mid=(i+j)/2;

    if(x[mid]==y){
        found=true;
        cout<<"founded in array number "<<mid;
    }
    else{
        if(x[mid]<y){
            i=mid+1;
        }
        else{
            j=mid-1;
        }
    }
}while((i<=j) && (not found));
```

```
    return 0;
}
```

Studi Kasus 4

```
/*
Nama    = Rio Sapta Samudera
NPM     = 140810180030
Kelas   = B
*/
#include <iostream>
using namespace std;

int main(){
    int i,j,ins,n=10,x[n]{1,20,12,41,10,32,41,22,92,70};

    for(i=1;i<n;i++){
        ins=x[i];
        j=i;

        while((j>0)&&(x[j-1]>ins)){
            x[j]=x[j-1];
            j=j-1;
```

```
        };
        x[j]=ins;
    }

    for(i=0;i<n;i++){
        cout<<x[i]<<endl;
    }

    return 0;
}
```

Studi Kasus 5

```
/*
Nama    = Rio Sapta Samudera
NPM     = 140810180030
Kelas   = B
*/
#include <iostream>
using namespace std;

int main(){
    int i,j,temp,n=10,x[n]{1,20,12,41,10,32,41,22,92,70};
```

```
    for(i=0;i<n;i++){
        for(j=i;j<n;j++){
            if(x[j]<x[i]){
                temp=x[i];
                x[i]=x[j];
                x[j]=temp;
            }
        }
    }

    for(i=0;i<n;i++){
        cout<<x[i]<<endl;
    }

    return 0;
}
```