**MODUL PRAKTIKUM 4**

**ANALISIS ALGORITMA**



**Disusun oleh :**

Rifaa’ Zalfaa’ Fakhriyyah 140810170031

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2019**

1. Merge Sort

Source code:

/\*\*\*\*

Nama Program : mengurutkan elemen menggunakan merge sort

Nama        : Rifaa' Zalfaa' Fakhriyyah

NPM         : 140810170031

\*\*\*/

#include <iostream>

#include<stdlib.h>

#include<stdio.h>

using namespace std;

void merge(int arr[], int left, int middle, int right)

{

int i, j, k;

int n1 = middle - left + 1;

int n2 = right - middle;

int kiri[n1], kanan[n2];

for (i = 0; i < n1; i++)

kiri[i] = arr[left + i];

for (j = 0; j < n2; j++)

kanan[j] = arr[middle + 1+ j];

i = 0;

j = 0;

k = left;

while (i < n1 && j < n2)

{

if (kiri[i] <= kanan[j])

{

arr[k] = kiri[i];

i++;

}

else

{

arr[k] = kanan[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = kiri[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = kanan[j];

j++;

k++;

}

}

void mergeSort(int arr[], int left, int right)

{

if (left < right)

{

int middle = left+(right-left)/2;

mergeSort(arr, left, middle);

mergeSort(arr, middle+1, right);

merge(arr, left, middle, right);

}

}

void printArray(int arr[], int size)

{

for (int i=0; i < size; i++)

cout<<arr[i]<<" ";

cout<<"\n";

}

int main()

{

    cout<<"=============================\n";

    cout<<"Merge Sort";

int arr[] = {12,10,18,20,16,133};

int arr\_size = sizeof(arr)/sizeof(arr[0]);

cout<<"Array nya \n";

printArray(arr, arr\_size);

mergeSort(arr, 0, arr\_size - 1);

cout<<"\nArray Setelah disorting \n";

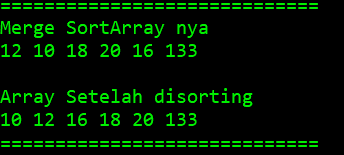
printArray(arr, arr\_size);

cout<<"=============================\n";

return 0;

}

Screenshot:



Kompleksitas Algoritma merge sort adalah O(n lg n).

1. Selection Sort

Source code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nama Program    : Selection sort

Nama            : Rifaa' Zalfaa' Fakhriyyah

NPM             : 140810170031

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include<conio.h>

using namespace std;

int data[100],data2[100];

int n;

void tukar(int a, int b)

{

    int t;

    t = data[b];

    data[b] = data[a];

    data[a] = t;

}

void selection\_sort()

{

    int pos,i,j;

    for(i=1;i<=n-1;i++)

    {

     pos = i;

     for(j = i+1;j<=n;j++)

     {

         if(data[j] < data[pos]) pos = j;

        }

if(pos != i) tukar(pos,i);

}

}

int main()

{

    cout << "\n=====================================";

    cout<<"\nMasukkan Jumlah Data : ";cin>>n;

    cout << "\n-------------------------------------" << endl;

    for(int i=1;i<=n;i++)

    {

        cout<<"Masukkan data ke-"<<i<<" : ";

        cin>>data[i];

        data2[i]=data[i];

    }

    selection\_sort();

    cout << "\n-------------------------------------" << endl;

    cout<<"Data Setelah di Sort : ";

    for(int i=1; i<=n; i++)

    {

       cout<<" "<<data[i];

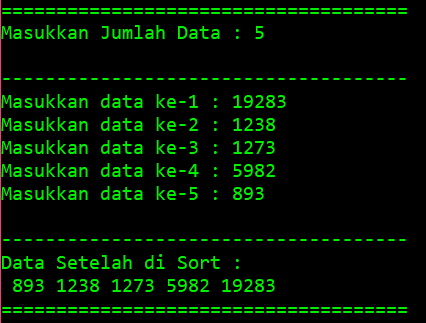
    }

    cout << "\n=====================================\n";

    getch();

}

Screenshot:



Menentukan T(n):

Oleh Karena itu:

Karena ,Maka

1. Insertion Sort

Source code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nama Program  : Insertion sort

Nama      : Rifaa' Zalfaa' Fakhriyyah

NPM       : 140810170031

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <conio.h>

using namespace std;

int data[100],data2[100],n;

void insertion\_sort()

{

  int temp,i,j;

  for(i=1;i<=n;i++){

   temp = data[i];

    j = i -1;

   while(data[j]>temp && j>=0){

      data[j+1] = data[j];

     j--;

   }

   data[j+1] = temp;

  }

}

int main()

{

  cout << "\n=================================="<<endl;

  cout<<"Masukkan Jumlah Data : "; cin>>n;

  cout<<endl;

  cout << "\n----------------------------------" << endl;

  for(int i=1;i<=n;i++)

  {

   cout<<"Masukkan data ke-"<<i<<" : ";

   cin>>data[i];

   data2[i]=data[i];

  }

  cout << "\n----------------------------------" << endl;

  insertion\_sort();

  cout<<"\nData Setelah di Sort : "<<endl;

  for(int i=1; i<=n; i++)

  {

   cout<<data[i]<<" ";

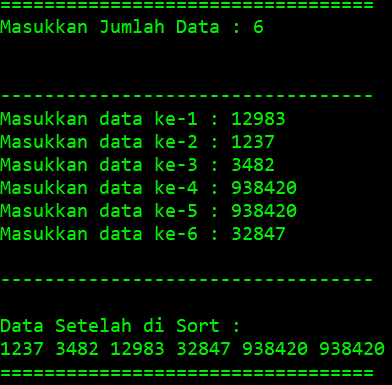
  }

  cout << "\n=================================="<<endl;

  getch();

}

Screenshot:



Menentukan T(n):

T(n) =

1. Bubble sort

Source code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nama Program    : Insertion sort

Nama            : Rifaa' Zalfaa' Fakhriyyah

NPM             : 140810170031

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <conio.h>

using namespace std;

int main(){

    int arr[100],n,temp;

    cout << "\n================================================"<<endl;

    cout<<"Massukan banyak elemen yang akan diinputkan : ";cin>>n;

    cout << "\n------------------------------------------------" << endl;

    for(int i=0;i<n;++i){

        cout<<"Masukkan Elemen ke-"<<i+1<<" : ";cin>>arr[i];

    }

    for(int i=1;i<n;i++){

        for(int j=0;j<(n-1);j++){

            if(arr[j]>arr[j+1]){

                temp=arr[j];

                arr[j]=arr[j+1];

                arr[j+1]=temp;

            }

        }

    }

    cout << "------------------------------------------------" << endl;

    cout<<"\nHasil dari Bubble Sort : "<<endl;

    for(int i=0;i<n;i++){

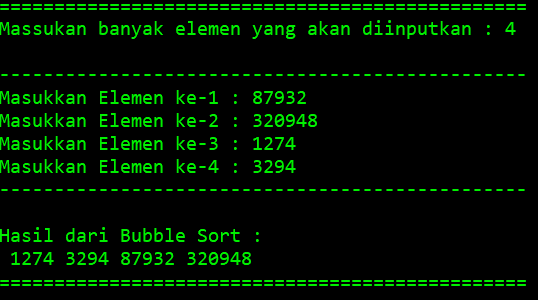
        cout<<" "<<arr[i];

    }

    cout << "\n================================================"<<endl;

}

Screenshot:



Menentukan T(n):