Algoritma & Struktur Data

**10. Selection dan insertion sort**

**Dosen Pengampu**

Dr. Tita Karlita S.Kom, M.Kom



**Disusun Oleh :**

Nama : M. Faza Nur Husain

Nrp : 3121550004

**D3 PJJ AK TEKNIK INFORMATIKA**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**TAHUN AKADEMIK 2021/2022**

1. Selection Sort

Source Code Program Selection Sort :

*#include* <stdio.h>

*#include* <sys/time.h>

void selectionSort(

    int array[]);

void printArray(

    int array[]);

struct timeval stop, start; *// membuat struck waktu harus membuat #include <sys/time.h> terlebih dahulu*

int main()

{

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

     puts("Metode Selection Sort");

     printArray(array);

}

void printArray(int array[])

{

     int size = 10;              *// panjang array mask. 10*

     printf("Array Awal  = [ "); *// isi dari array 1 - 4 akan dipanggil kedalam sini*

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

     {

          printf("%d ", array[i]);

     }

     printf("]\n");

     gettimeofday(&start, NULL); *// waktu mulai jalannya sorting*

     selectionSort(array);       *// setelah tercetak akan memanggil array dari selection sort*

}

void selectionSort(int array[])

{

     int size = 10;

     int perbandingan = 0;

     int penukaran = 0;

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

     {

          int min = i;

*for* (int j = i + 1; j < size; j++)

          {

*if* (array[j] < array[min])

               {

                    min = j;

               }

               perbandingan++; *// nilai perbandingan akan di hitung n \* (n-1) / 2*

          }

          int temp = array[i];

*if* (temp != array[min])

          {

               penukaran++; *// menghitung berapa kali penukaran telah dikerjakan*

               array[i] = array[min];

               array[min] = temp;

          }

     }

     printf("Array hasil = [ ");

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

     {

          printf("%d ", array[i]);

     }

     printf("]\n");

     gettimeofday(&stop, NULL); *// waktu mulai berhentinya jalan sorting*

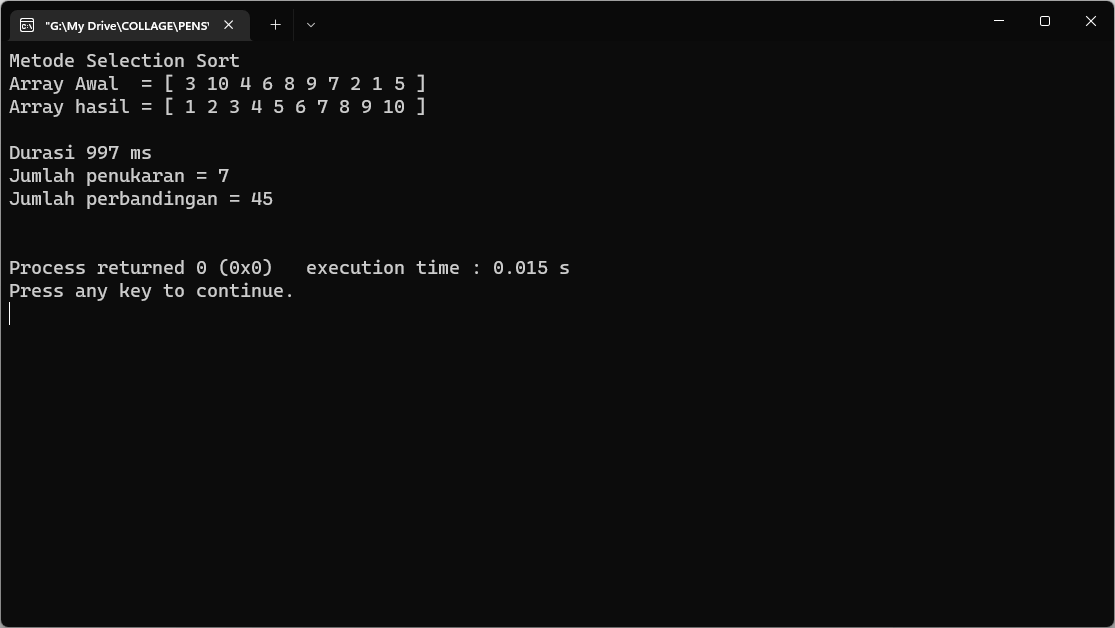
     printf("\nDurasi %lu ms\n", (stop.tv\_sec - start.tv\_sec) \* 1000000 + stop.tv\_usec - start.tv\_usec);

     printf("Jumlah penukaran = %d\n", penukaran);

     printf("Jumlah perbandingan = %d\n\n", perbandingan);

}

Output Source Code Selection Sort :



1. Insertion Sort

Source Code Program Insertion Sort :

*#include* <stdio.h>

*#include* <sys/time.h>

void insertionSort(

    int array[]);

void printArray(

    int array[]);

int main()

{

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

     puts("Metode Insertion Sort");

     printArray(array);

}

struct timeval stop, start; *// membuat struck waktu harus membuat #include <sys/time.h> terlebih dahulu*

void printArray(int array[])

{

     int size = 10;

     printf("Array Awal  = [ "); *// data array 1 - 4 akan dipanggil kedalam sini*

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

     {

          printf("%d ", array[i]);

     }

     printf("]\n");

     gettimeofday(&start, NULL); *// waktu mulai jalannya sorting*

     insertionSort(array);       *// selanjutkan akan diurutkan dengan insertionsort*

}

void insertionSort(int array[])

{

     int size = 10, perbandingan = 0, penyisipan = 0, pergeseran = 0;

     int i, j, x;

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

     {

          int min = i;

*for* (i = 1; i < size; i++)

          {

               x = array[i]; *// Assign elemen array indeks i ke x*

               j = i - 1;

*// Inisialisasi j utk pembandingan*

*// bandingkan elemen array pd indeks j dgn x*

*// if j >= 0 dan elemen indeks j > x*

               perbandingan++; *// menghitung perbandingan n-1*

*while* (x < array[j])

               {

                    array[j + 1] = array[j]; *// pindahkan elemen tsb ke 1 posisi berikutnya*

                    j = j - 1;               *// go to next lower element*

                    pergeseran++;            *// menghitung berapakali terjadi pergeseran*

               }                             *// Lanjutkan sampai A[j] not > x*

*if* (array[j + 1] != x)

               {

                    array[j + 1] = x; *// assign temp kembali ke array*

                    penyisipan++;     *// menghitung berapakali terjadi penyisipan*

               }

          }

     }

     printf("Array hasil = [ ");

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

     {

          printf("%d ", array[i]);

     }

     printf("]\n");

     gettimeofday(&stop, NULL); *// waktu mulai berhentinya jalan sorting*

     printf("\nDurasi %lu ms\n", (stop.tv\_sec - start.tv\_sec) \* 1000000 + stop.tv\_usec - start.tv\_usec);

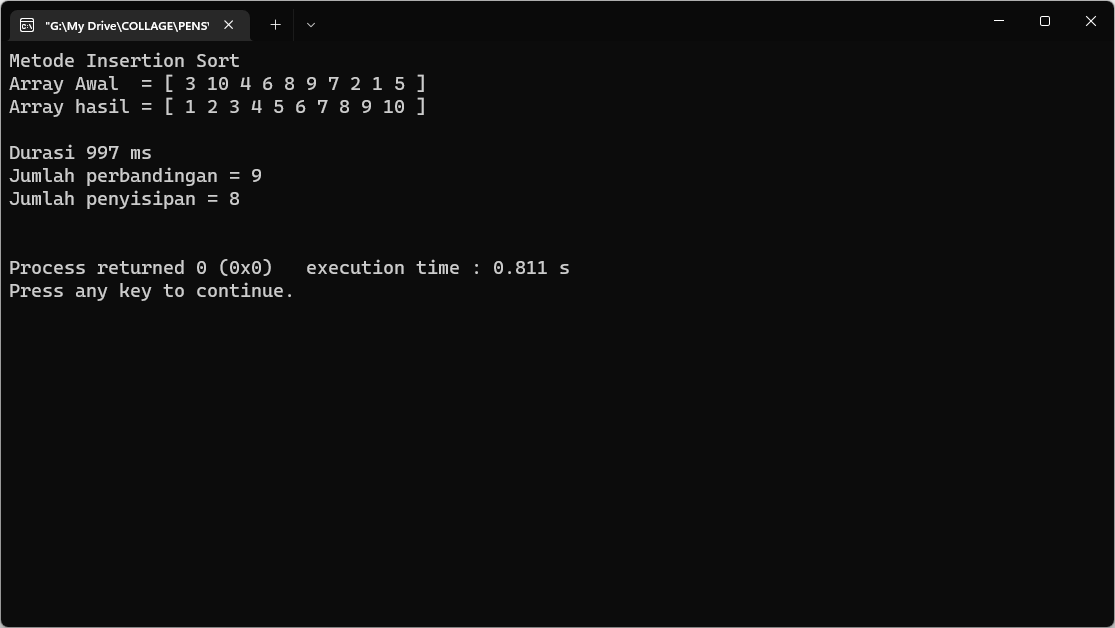
     printf("Jumlah perbandingan = %d\n", perbandingan);

*// printf("Jumlah pergeseran = %d\n", pergeseran);*

     printf("Jumlah penyisipan = %d\n\n", penyisipan);

}

Output Source Code Insertion Sort :



1. Analisa terhadap no 1 dan 2. terkait jumlah perbandingan, jumlah penukaran dan waktu komputasi.

|  |  |  |
| --- | --- | --- |
|  | Selection Sort | Insertion Sort |
| Perbandingan | 45 | 9 |
| Penukaran | 7 | 8 |