LAPORAN RESMI
PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

SEARCHING (Sequential & Binary)

Nama : Aqilah Akmalia Dewi

Kelas : 1 D4 IT B

NRP : 3120600046

PROGRAM STUDI D4 TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2021

DAFTAR ISI

1. LISTING PROGRAM

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <sys/time.h>
#define MAX 100000

void menu();
void binary(int []);
void sequential(int []);
void shell_sort(int []);
void generate(int x[]);
void copy(int [], int[]);
void swap(int *a, int *b);

int n, pil;
int data[MAX], data2[MAX];
struct timeval start, stop;
double secs = 0;

int main(){
    puts("Menu Searching - Data Int\n");
    printf("Masukkan Banyak Data (maks 100000) : ");
    scanf("%d", &n);
    srand(time(NULL));
    generate(data);
    copy(data, data2);
    do{
        fflush(stdin);
        copy(data2, data);
        //printf("Array : ");
        //tampil(data);
        puts("");
        menu();
        printf("Durasi : %6.2f milliseconds", secs);
        puts("\n----------------------------------------");
    }while(pil!=4);
    return 0;

}

void menu(){
    printf("Menu Searching\n");
    printf("1. Sequential Search (unsorted)\n");
    printf("2. Sequential Search (sorted)\n");
    printf("3. Binary Search (sorted)\n");
    printf("4. Keluar\n");
    printf("Pilihan Anda : ");
    scanf("%d", &pil);

    switch(pil)
    {
        case 1:
            fflush(stdin);
            puts("");
            gettimeofday(&start, NULL);
            sequential(data);
            gettimeofday(&stop, NULL);
            secs = (double)(stop.tv_usec - start.tv_usec) / 1000000 +
(double)(stop.tv_sec - start.tv_sec);
```

```
            break;
        case 2:
            fflush(stdin);
            puts("");
            shell_sort(data);
            //printf("Array terurut : ");
            //tampil(data);
            //puts("");
            gettimeofday(&start, NULL);
            sequential(data);
            gettimeofday(&stop, NULL);
            secs = (double)(stop.tv_usec - start.tv_usec) / 1000000 +
(double)(stop.tv_sec - start.tv_sec);
            break;
        case 3:
            fflush(stdin);
            puts("");
            shell_sort(data);
            //printf("Array terurut: ");
            //tampil(data);
            //puts("");
            gettimeofday(&start, NULL);
            binary(data);
            gettimeofday(&stop, NULL);
            secs = (double)(stop.tv_usec - start.tv_usec) / 1000000 +
(double)(stop.tv_sec - start.tv_sec);
            break;
        default:
            exit(0);
    }
}

void generate(int x[]){
      int i;

      for(i=0; i<n; i++)
            x[i] = rand()/1000;
}

void copy(int data[], int data2[]){
    int i;

    for(i=0; i<n; i++)
        data2[i] = data[i];
}

void swap(int *a, int *b){
    int c;

    c = *a;
    *a = *b;
    *b = c;
}

void binary(int data[]){
    int L, R, m, key, found=0, banding=0;

    printf("Input data yang ingin dicari : ");
    scanf("%d", &key);

    L = 0;
```

```
    R = n - 1;

    while((L <= R) && (found==0)) {
        m = (L + R) / 2;
        banding++;
        if (data[m] == key)
            found++;
        else if (data[m] < key){
            L = m + 1;
        }else{
            R = m - 1;
        }
    }

    if(found==1)
        printf("%d berada di indeks ke %d\n", key, m+1);
    else
        printf("d tidak ada\n", key);
    printf("Jumlah Perbandingan : %d\n", banding);
}

void sequential(int data[]){
    int key, i=0, banding=0, found=0;

    printf("Input data yang ingin dicari : ");
    scanf("%d", &key);

    i = 0;
    while((i<n) && (found==0)){
        banding++;
        if (data[i] == key)
            found++;
        else{
            i++;
        }
    }

    if (found==1)
        printf("%d berada di indeks ke %d\n", key, i+1);
    else
        printf("%d tidak ada\n", key);
    printf("Jumlah Perbandingan : %d\n", banding);
}

void shell_sort(int data[]){
    int i, j, jarak, did_swap;

    jarak = n;
    did_swap = 1;

    while(jarak>1){
        jarak = jarak/2;
        did_swap = 1;
        while(did_swap==1){
            did_swap = 0;
            i = 0;
            while(i<(n-jarak)){
                if(data[i]>data[i+jarak]){
                    swap(&data[i],&data[i+jarak]);
                    did_swap = 1;
                }
```

```
                i++;
            }
        }
    }
}

void tampil(int data[]){
    for(int i=0; i<n; i++){
            printf("%d ", data[i]);
      }
}
```

2. OUTPUT

a. Data 25.000

```
"C:\Users\acer\OneDrive\Documents\SEMESTER 2\A
Menu Searching - Data Int

Masukkan Banyak Data (maks 100000) : 25000

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 1

Input data yang ingin dicari : 9
9 berada di indeks ke 46
Jumlah Perbandingan : 46
Durasi :   0.37 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 2

Input data yang ingin dicari : 9
9 berada di indeks ke 6816
Jumlah Perbandingan : 6816
Durasi :   0.83 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 3

Input data yang ingin dicari : 9
9 berada di indeks ke 7031
Jumlah Perbandingan : 5
Durasi :   0.50 milliseconds
----------------------------------------
```

b. Data 50.000

```
"C:\Users\acer\OneDrive\Documents\SEMESTER 2\ASD
Menu Searching - Data Int

Masukkan Banyak Data (maks 100000) : 50000

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 1

Input data yang ingin dicari : 4
4 berada di indeks ke 6
Jumlah Perbandingan : 6
Durasi :   0.63 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 2

Input data yang ingin dicari : 4
4 berada di indeks ke 6255
Jumlah Perbandingan : 6255
Durasi :   0.67 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 3

Input data yang ingin dicari : 4
4 berada di indeks ke 7031
Jumlah Perbandingan : 6
Durasi :   0.50 milliseconds
----------------------------------------
```

c. Data 75.000

```
Select "C:\Users\acer\OneDrive\Documents\SEMESTER
Menu Searching - Data Int

Masukkan Banyak Data (maks 100000) : 75000

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 1

Input data yang ingin dicari : 5
5 berada di indeks ke 18
Jumlah Perbandingan : 18
Durasi :   0.94 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 2

Input data yang ingin dicari : 5
5 berada di indeks ke 11432
Jumlah Perbandingan : 11432
Durasi :   0.82 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 3

Input data yang ingin dicari : 5
5 berada di indeks ke 11718
Jumlah Perbandingan : 5
Durasi :   0.49 milliseconds
----------------------------------------
```

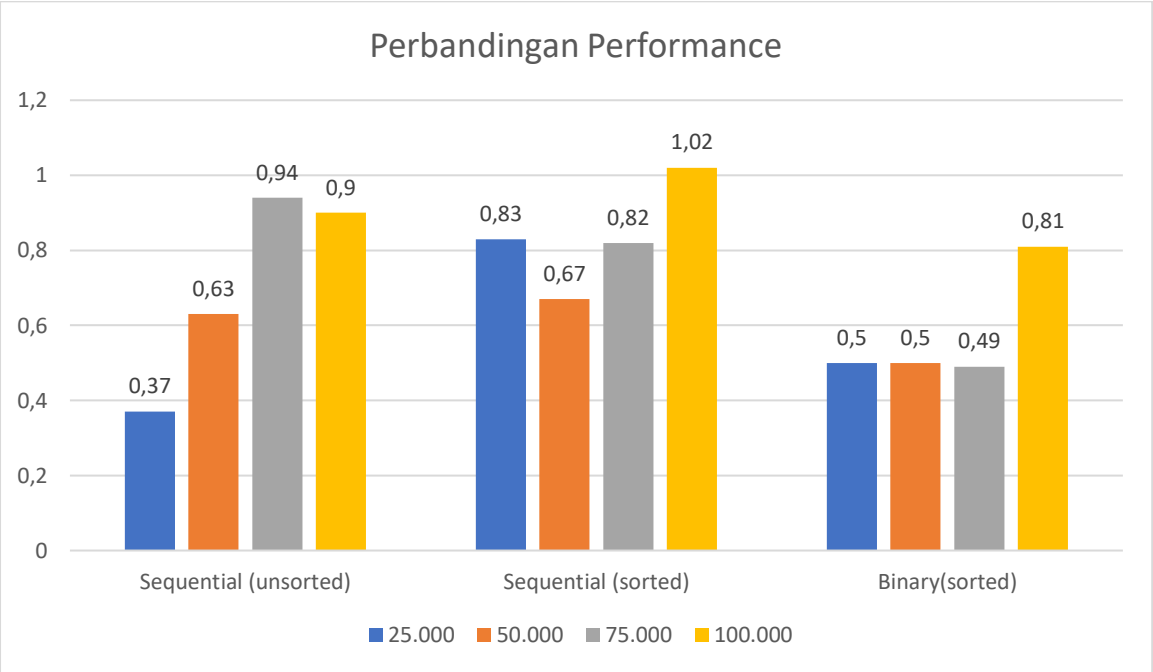
d. Data 100.000

```
"C:\Users\acer\OneDrive\Documents\SEMESTER 2\ASD
Menu Searching - Data Int

Masukkan Banyak Data (maks 100000) : 100000

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 1

Input data yang ingin dicari : 6
6 berada di indeks ke 80
Jumlah Perbandingan : 80
Durasi :   0.90 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 2

Input data yang ingin dicari : 6
6 berada di indeks ke 18179
Jumlah Perbandingan : 18179
Durasi :   1.02 milliseconds
----------------------------------------

Menu Searching
1. Sequential Search (unsorted)
2. Sequential Search (sorted)
3. Binary Search (sorted)
4. Keluar
Pilihan Anda : 3

Input data yang ingin dicari : 6
6 berada di indeks ke 18750
Jumlah Perbandingan : 4
Durasi :   0.81 milliseconds
----------------------------------------
```

3. TABEL DURASI WAKTU

Metode Searching	25.000	50.000	75.000	100.000
Sequential (unsorted)	0.37	0.63	0.94	0.90
Sequential (sorted)	0.83	0.67	0.82	1.02
Binary (sorted)	0.50	0.50	0.49	0.81

4. DIAGRAM BATANG



6. KESIMPULAN

Kesimpulan yang dapat saya ambil dari hasil diagram batang di atas, Binary (sorted) merupakan metode searching yang paling efisien dibandingkan yang lain. Waktu komputasinya cenderung singkat. Mesikpun begitu, binary search ini hanya dapat dilakukan pada data yang sudah terurut, jadi kita harus menambahkan sorting terlebih dahulu sebelum memanggil fungsinya. Konsep dasar metode ini adalah membagi 2 jumlah elemennya, dan menentukan apakah data yang berada pada elemen paling tengah bernilai sama, lebih dari atau kurang dari nilai data yang akan dicari. Jika bernilai sama, maka langsung data yang dicari ditemukan. Hal tersebut yang mungkin berdampak pada keefisienan binary search yang lebih unggul. Berbeda dengan sequential search, ia bisa melakukan searching pada data terurut maupun tidak terurut. Karena prosesnya itu membandingkan setiap elemen array dari awal sampai akhir secara berurutan sampai elemen yang dicari ditemukan. Hal tersebut menyebabkan sequential search membutuhkan waktu yang lebih lama, jika data yang dicari berada pada posisi akhir. Tapi jika data berada pada posisi depan, tentu waktunya akan lebih singkat.