

Laporan Resmi
Praktikum Algoritma dan Struktur Data
Sequence Searching and Binary Searching



Dr. Tita Karlita S.Kom, M.Kom

Nama : Marits Ikmal Yasin
Kelas : 1D4 IT B
NRP : 3121600047

1. Membuat menu untuk membandingkan kecepatan antara sequence search dengan binary search.

```
Kode      :
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include <sysinfoapi.h>
#define MAKS 100000

void urutkan(int[], int);
void tampilkan(int[], int);
void sequence(int[], int, int);
void binary(int[], int, int);
void isi(int[], int[], int);
void menu();
// long gettimeofday();

int main()
{
    int panjang, pilihan, key,urut = 0;
    char jwb;
    // clock_t start, end;
    struct timeval stop, start;

    printf("Masukkan Banyak Data : ");
    scanf("%d", &panjang);
    int array[panjang], sort[panjang];

    isi(array, sort, panjang);
    tampilkan(array, panjang);
    do
    {
        fflush(stdin);
        menu();
        printf("Masukkan pilihan Anda : ");
        scanf("%d", &pilihan);
        if (pilihan != 1 && urut == 0)
        {
            urut++;
            urutkan(sort, panjang);
        }
        printf("Masukkan data yang ingin dicari : ");
        scanf("%d", &key);
        gettimeofday(&start, NULL);
        // long start = gettimeofday();
        // // start = clock();
        // // long long int begin = GetTickCount64();
        // sleep(0.10);
        switch (pilihan)
        {
            case 1:
                sequence(array, panjang, key);
```

```

        break;
    case 2:
        sequence(sort, panjang, key);
        break;
    case 3:
        binary(sort, panjang, key);
        break;
    case 4:
        exit(0);
    default:
        puts("Input Yang Benar");
    }
    gettimeofday(&stop, NULL);
    printf("took %lu us\n", ((stop.tv_sec - start.tv_sec) * 1000000 + stop.tv_usec -
start.tv_usec));

    // end = clock();
    // double extime = (double)(end - start) * 1000 / CLOCKS_PER_SEC;
    // printf("Searching took %f milliseconds to execute \n", extime);

    // long long int end = GetTickCount64();
    // double elapsed = (end - begin) * 1e-3;
    // printf("Time measured: %.3f seconds.\n", elapsed);

    // long end = gettimeofday();
    // printf("Time Measured : %ld\n", end - start);
    fflush(stdin);
    printf("Mau lagi (y/t)? ");
    scanf("%c", &jwb);
} while (jwb == 'y' || jwb == 'Y');
// total_t = (double)(end_t - start_t) / CLOCKS_PER_SEC;
// printf("Total time taken by CPU: %f\n", total_t);
return 0;
}

void menu()
{
    puts("1. Sequential Search Unsorted Array");
    puts("2. Sequential Search Sorted Array");
    puts("3. Binary Search Sorted Array");
    puts("4. Keluar");
}

void isi(int array[], int sort[], int panjang)
{
    for (int i = 0; i < panjang; i++)
    {
        array[i] = sort[i] = rand();
    }
}

void tampilkan(int array[], int panjang)
{
    for (int i = 0; i < panjang; i++)

```

```
{  
    printf("%d ", array[i]);  
}  
puts(" ");  
}
```

```
void sequence(int array[], int panjang, int key)  
{  
    int valid = 0, perbandingan = 0;  
  
    for (int i = 0; i < panjang; i++)  
    {  
        usleep(1000);  
        perbandingan++;  
        if (key == array[i])  
        {  
            printf("%d ada di dalam Array indeks ke %d\n", key, i);  
            valid = 1;  
            break;  
        }  
    }  
    if (!valid)  
    {  
        puts("Data tidak ditemukan di dalam Array");  
    }  
    printf("Perbandingan : %d\n", perbandingan);  
}
```

```
void binary(int array[], int panjang, int key)  
{  
    int kiri = 0, kanan = panjang - 1, tengah, valid = 0, perbandingan = 0;  
  
    while (kiri <= kanan)  
    {  
        usleep(1000);  
        perbandingan++;  
        tengah = (kiri + kanan) / 2;  
        if (array[tengah] == key)  
        {  
            printf("%d ada di dalam Array indeks ke %d\n", key, tengah);  
            valid = 1;  
            break;  
        }  
        else if (array[tengah] < key)  
        {  
            kiri = tengah + 1;  
        }  
        else  
        {  
            kanan = tengah - 1;  
        }  
    }  
    if (!valid)  
    {
```

```

        puts("Data tidak ditemukan di dalam Array");
    }
    printf("Perbandingan : %d\n", perbandingan);
}

```

```

// bubble sort
void urutkan(int sort[], int panjang)
{
    int cek, akhir = panjang, temp;

    for (int i = 0; i < panjang; i++)
    {
        cek = 0;
        akhir--;
        for (int j = 0; j < akhir; j++)
        {
            if (sort[j] > sort[j + 1])
            {
                temp = sort[j];
                sort[j] = sort[j + 1];
                sort[j + 1] = temp;
            }
            else
            {
                cek++;
                if (cek == panjang - 1)
                    puts("Data telah diurutkan");
            }
        }
    }
    tampilkan(sort, panjang);
}

```

```

// long gettimeofday()
// {
//     struct timeval start, end;
//     gettimeofday(&start, NULL);
//     return start.tv_sec * 1000 + start.tv_usec;
// }

```

Output :

Pilihan 1 dengan memilih indeks 0 dan 99.

```

Masukkan Banyak Data : 100
41 18467 6334 26500 19169 15724 11478 29358 26962 24464 5705 28145 23281 16827 9961 491 2995 11942 4827 5436 32391 14604 3902
153 292 12382 17421 18716 19718 19895 5447 21726 14771 11538 1869 19912 25667 26299 17035 9894 28703 23811 31322 30333 17673
4664 15141 7711 28253 6868 25547 27644 32662 32757 20037 12859 8723 9741 27529 778 12316 3035 22190 1842 288 30106 9040 8942
19264 22648 27446 23805 15890 6729 24370 15350 15006 31101 24393 3548 19629 12623 24084 19954 18756 11840 4966 7376 13931 26
308 16944 32439 24626 11323 5537 21538 16118 2082 22929 16541
1. Sequential Search Unsorted Array
2. Sequential Search Sorted Array
3. Binary Search Sorted Array
4. Keluar
Masukkan pilihan Anda : 1
Masukkan data yang ingin dicari : 41
41 ada di dalam Array indeks ke 0
Perbandingan : 1
took 16984 us
Mau lagi (y/t)? y
1. Sequential Search Unsorted Array
2. Sequential Search Sorted Array
3. Binary Search Sorted Array
4. Keluar
Masukkan pilihan Anda : 1
Masukkan data yang ingin dicari : 16541
16541 ada di dalam Array indeks ke 99
Perbandingan : 100
took 1056808 us
Mau lagi (y/t)?

```

Pilihan 2 dengan memilih indeks 0 dan 99.

```
Masukkan Banyak Data : 100
41 18467 6334 26500 19169 15724 11478 29358 26962 24464 5705 28145 23281 16827 9961 491 2995 11942 4827 5436 32391 14604 3902 153 292
12382 17421 18716 19718 19895 5447 21726 14771 11538 1869 19912 25667 26299 17035 9894 28703 23811 31322 30333 17673 4664 15141 7711
28253 6868 25547 27644 32662 32757 20037 12859 8723 9741 27529 778 12316 3035 22190 1842 288 30106 9040 8942 19264 22648 27446 23805
15890 6729 24370 15350 15006 31101 24393 3548 19629 12623 24084 19954 18756 11840 4966 7376 13931 26308 16944 32439 24626 11323 5537
21538 16118 2082 22929 16541
1. Sequential Search Unsorted Array
2. Sequential Search Sorted Array
3. Binary Search Sorted Array
4. Keluar
Masukkan pilihan Anda : 2
41 153 288 292 491 778 1842 1869 2082 2995 3035 3548 3902 4664 4827 4966 5436 5447 5537 5705 6334 6729 6868 7376 7711 8723 8942 9040
9741 9894 9961 11323 11478 11538 11840 11942 12316 12382 12623 12859 13931 14604 14771 15006 15141 15350 15724 15890 16118 16541 1682
7 16944 17035 17421 17673 18467 18716 18756 19169 19264 19629 19718 19895 19912 19954 20037 21538 21726 22190 22648 22929 23281 23805
23811 24084 24370 24393 24464 24626 25547 25667 26299 26308 26500 26962 27446 27529 27644 28145 28253 28703 29358 30106 30333 31101
31322 32391 32439 32662 32757
Masukkan data yang ingin dicari : 41
41 ada di dalam Array indeks ke 0
Perbandingan : 1
took 13510 us
Mau lagi (y/t)? y
1. Sequential Search Unsorted Array
2. Sequential Search Sorted Array
3. Binary Search Sorted Array
4. Keluar
Masukkan pilihan Anda : 2
Masukkan data yang ingin dicari : 32757
32757 ada di dalam Array indeks ke 99
Perbandingan : 100
took 1562118 us
Mau lagi (y/t)? |
```

Pilihan 3 dengan memilih indeks 0 dan 99.

```
Masukkan Banyak Data : 100
41 18467 6334 26500 19169 15724 11478 29358 26962 24464 5705 28145 23281 16827 9961 491 2995 11942 4827 5436 32391 14604 3902 153 292 12382 1742
1 18716 19718 19895 5447 21726 14771 11538 1869 19912 25667 26299 17035 9894 28703 23811 31322 30333 17673 4664 15141 7711 28253 6868 25547 2764
4 32662 32757 20037 12859 8723 9741 27529 778 12316 3035 22190 1842 288 30106 9040 8942 19264 22648 27446 23805 15890 6729 24370 15350 15006 311
01 24393 3548 19629 12623 24084 19954 18756 11840 4966 7376 13931 26308 16944 32439 24626 11323 5537 21538 16118 2082 22929 16541
1. Sequential Search Unsorted Array
2. Sequential Search Sorted Array
3. Binary Search Sorted Array
4. Keluar
Masukkan pilihan Anda : 3
41 153 288 292 491 778 1842 1869 2082 2995 3035 3548 3902 4664 4827 4966 5436 5447 5537 5705 6334 6729 6868 7376 7711 8723 8942 9040 9741 9894 9
961 11323 11478 11538 11840 11942 12316 12382 12623 12859 13931 14604 14771 15006 15141 15350 15724 15890 16118 16541 16827 16944 17035 17421 17
673 18467 18716 18756 19169 19264 19629 19718 19895 19912 19954 20037 21538 21726 22190 22648 22929 23281 23805 23811 24084 24370 24393 24464 24
626 25547 25667 26299 26308 26500 26962 27446 27529 27644 28145 28253 28703 29358 30106 30333 31101 31322 32391 32439 32662 32757
Masukkan data yang ingin dicari : 41
41 ada di dalam Array indeks ke 0
Perbandingan : 6
took 106488 us
Mau lagi (y/t)? y
1. Sequential Search Unsorted Array
2. Sequential Search Sorted Array
3. Binary Search Sorted Array
4. Keluar
Masukkan pilihan Anda : 3
Masukkan data yang ingin dicari : 16827
16827 ada di dalam Array indeks ke 50
Perbandingan : 1
took 9554 us
Mau lagi (y/t)? |
```

Analisa :
Waktu yang diperlukan dalam menggunakan sequence search cenderung lebih lama. Sedangkan saat kita menggunakan binary search lebih cepat. Hal itu disebabkan karena saat menggunakan sequence search kita perlu mengecek data satu-persatu sesuai urutan array. Sedangkan binary search melakukan perbandingan dengan data yang berada indeks di tengah. Oleh karena itu binary search lebih cepat daripada sequence search.