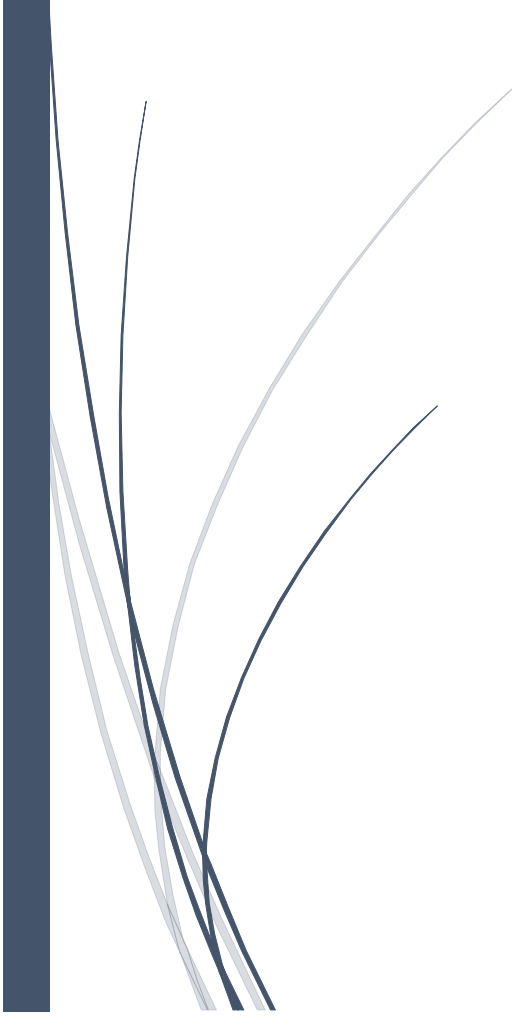
2022/2023

SEARCHING

MODUL #13



Asisten Informasi & Struktur Data 2022/2023

A. Tujuan

1. Pratikan dapat memahami metode searching khususnya sequential search dan binary search.

B. Pengertian Searching

Searching merupakan suatu proses pencarian data dalam sekumpulan data dengan menggunakan kunci data yang dapat perbandingkan. Terdapat hal-hal yang harus diperhatikan dalam searching.

a. Data mempunyai kunci atau sesautu yang dapat diperbandingkan dengan data yang lain, bisa dibilang data yang unik.
b. Data tersimpan dalam suatu array yang sudah terurut(sorted) atau belum.
c. Data yang dicari pada array dapat dicari atau tidak.

C. Jenis-jenis Searching

Pada modul ini hanya akan membahas 2 jenis metode searching yaitu sequential atau linear search dan Binary search. Kedua metode ini sering digunakan dan mudah dipahami. Untuk metode searching lain, kalian bisa mempelajarinya dalam pada link ini:

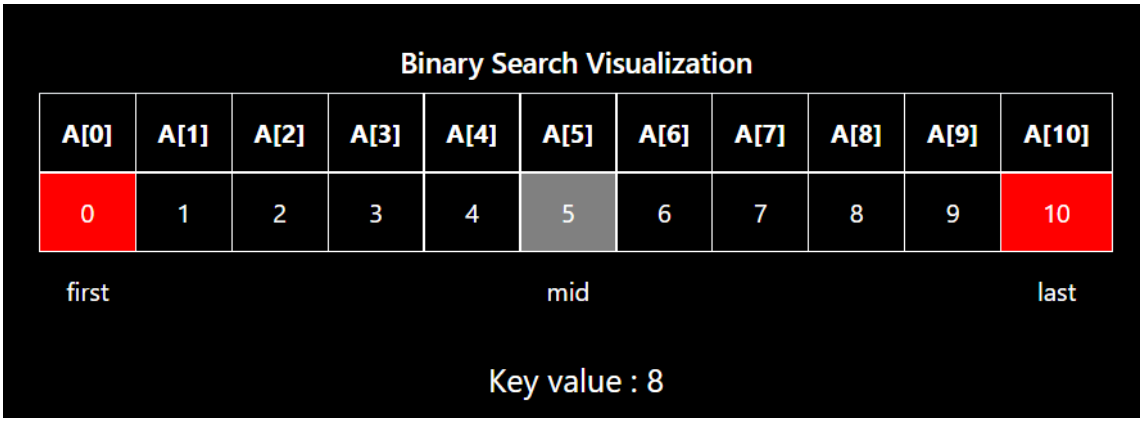
- https://www.analyticsvidhya.com/blog/2021/09/searching-in-data-structure-different-search-methods-explained/#:~:text=What%20is%20Searching%20in%20Data,list%2C%20graph%2C%20or%20tree.
- https://www.geeksforgeeks.org/searching-algorithms/

a. Sequential/Linear Search

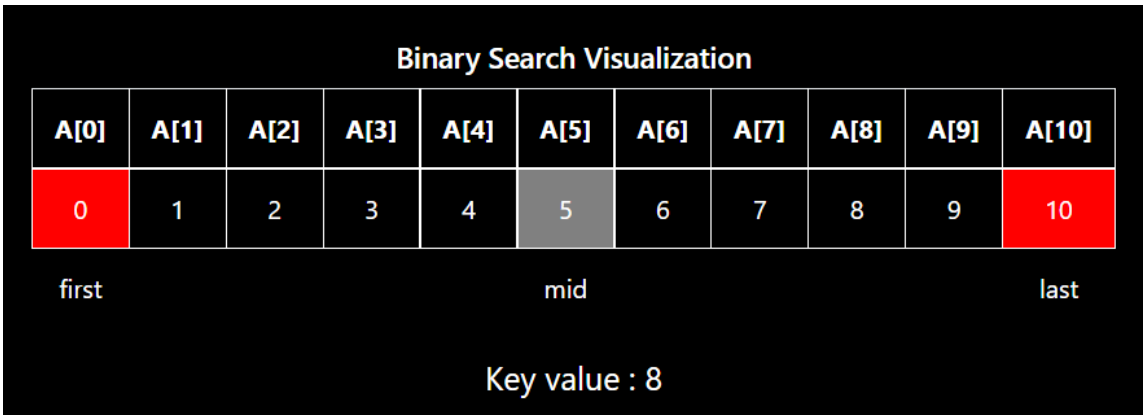
Sebelumnya kita sudah sering menggunakan metode searching jenis ini, yaitu menggunakan fungsi untuk mencari index pada suatu array. Jenis search ini merupakan jenis search paling sederhana dan mudah digunakan. Dengan sequential search biasanya melakukan search dari elemen pertama sampai elemen terakhir atau dari elemen terakhir sampai elemen pertama dari sebuah array.

b. Binary Search

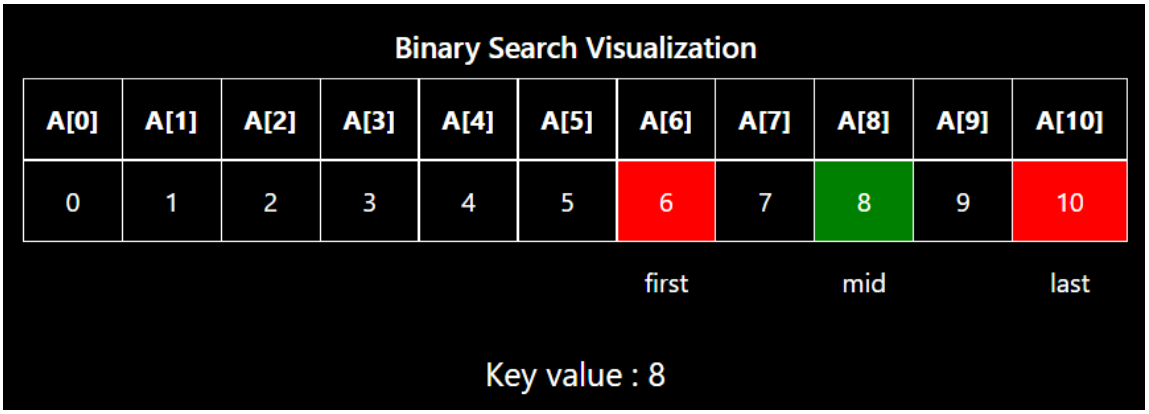
Metode binary search lebih efinsien dari pada menggunakan sequensial search yang membandingkan satu persatu elemen dalam sebuah array. Menggunakan binary search diwajibkan mengurutkan(sorting) data dalam array terlebih dahulu sebelum mencari data yang dibutuhkan. Berikut ini merupakan langka pengerjaan binary search:

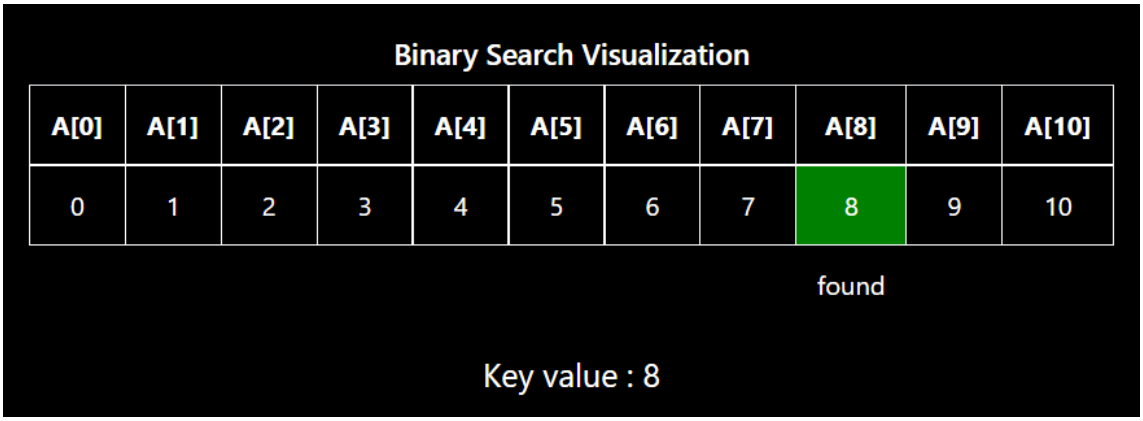


1. Pertama-tama pastikan array sudah terurut, untuk pengurutan data bisa dilihat di materi modul 12 yaitu sorting.



2. Selanjutnya menetapkan index pertama(first), index terkahir(last) dan index tengah(mid). Penetapan ini bertujuan untuk membagi array menjadi sub-bagian kiri yaitu nilai yang berada dikiri index tengah dan sub-bagian kanan yaitu nilai yang berada di kanan dari index tengah.





3. Kemudian bandingkan nilai kunci yaitu 8 dengan nilai index tengah(mid) apakah sama, jika tidak bandingkan nilai yang dicari apakah lebih besar dari nilai index tengah maka lakukan ulang Langkah 2 dengan hanya menggunakan sub bagian kanan.

4. Namun jika nilai yang dicari lebih kecil dari nilai index tengah maka lakukan Langkah 2 dengan menggunakan hanya sub-bagian kiri.

5. Searching akan berhenti jika berhasil menemukan index data yang dicari, selain itu searching juga akan berhenti jika tidak menemukan index datanya.

Guided

Kumpul dengan format **GD13_Kelas_5 digit npm.** Salah format -10

Header.h

```h
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>

#define N 10

typedef int infoType;
typedef infoType keyType;
typedef infoType arrayType[N];

int isEmpty(arrayType a);
int isFull(arrayType a);
void createEmpty(arrayType a);
void swapElement(keyType *first, keyType *second);
void bubbleSort(arrayType a);
int binarySearch(arrayType a, int left, int right, infoType find);
int sequentialSearch(arrayType a, infoType find);
void printArray(arrayType a);
void copyArray(arrayType a, arrayType b);
int cekUnique(arrayType a, infoType input);
void insertRandom(arrayType a);
```

Source.c

```c
#include "header.h"

int isEmpty(arrayType a){
    return (a[0] == 0);
}

int isFull(arrayType a){
    return (a[N-1] != 0);
}

void createEmpty(arrayType a){
    int i;
    for(i=0;i<N;i++){
        (a)[i] = 0;
    }
}

void swapElement(keyType *first, keyType *second)
{
    keyType temp;
    temp = (*first);
    (*first) = (*second);
    (*second) = temp;
}

void bubbleSort(arrayType a)
{
    int i, j;
    for(i=0;i<N;i++){
        for(j=i+1;j<N;j++){
            if((a)[i] > (a)[j]){
                swapElement(&(a)[i], &(a)[j]);
            }
        }
    }
}

int binarySearch(arrayType a, int left, int right, infoType find)
{
    if(right >= left){
        int mid = (left+right)/2;

        if(find == a[mid]){
            return mid;
        }

        if(find < a[mid]){
            return binarySearch(a, left, mid-1, find);
        }

        return binarySearch(a, mid+1, right, find);
    }

    return -1;
}
```

```c
int sequentialSearch(arrayType a, infoType find){

    int i;
    for(i=0;i<N;i++){
        if(a[i] == find){
            return i;
        }
    }
    return -1;
}

void printArray(arrayType a)
{
    int i;
    for(i=0;i<N;i++){
        printf("%d ", a[i]);
    }
}

int cekUnique(arrayType a, infoType input)
{
    int i;
    for(i=0;i<N;i++){
        if(a[i]==input){
            return 0;
        }
    }
    return 1;
}

void copyArray(arrayType a, arrayType b)
{
    int i;

    for(i=0;i<N;i++){
        b[i] = a[i];
    }
}

void insertRandom(arrayType a)
{
    int i, num;
    srand((unsigned) time(NULL));
    for(i=0;i<N;i++){
        num = rand() % 70 + 1; //men generate angka dari 1-70
        if(cekUnique(a, num))
            (a)[i] = num;
        else i--;
    }
}
```

Main.c

```c
#include "header.h"

int main(int argc, char *argv[]) {

    arrayType array, sortArray;
    int menu;
    int find, found;

    createEmpty(array);

    do{
        system("cls");
        system("color F0");
        printf("\n\n");
        printf("\t-- GUIDED SEARCHING ---\n");
        printf("\n\t [1] Generate Number");
        printf("\n\t [2] Sequential Search");
        printf("\n\t [3] Binary Search");
        printf("\n\t [0] End Program");
        printf("\n\t >>>> "); scanf("%d",&menu);

        switch(menu){
            case 1:
                // generate data array

                if(isFull(array)) createEmpty(array);


                insertRandom(array);
                printf("\n\tIsi Array Generate: ");
                printArray(array);
                break;

            case 2:
                // searching data dengan cara sequential search

                if(isEmpty(array)){
                    printf("\n\tArray Kosong [!]");
                    break;
                }

                printf("\n\tData array : ");
                printArray(array);
```

```
                printf("\n\tCari : "); scanf("%d",&find);
                found = sequentialSearch(array, find);

                if(found == -1) printf("\n\tData tidak ditemukan [!]");
                else printf("\n\tData %d ditemukan di indeks ke-%d", array[found], found);
            break;

            case 3:
                // searching data yang telah di shorting dengan binary search

                if(isEmpty(array)){
                    printf("\n\tArray Kosong [!]");
                    break;
                }

                printf("\n\tSebelum Sorting : "); printArray(array);
                copyArray(array, sortArray); // mengcopy data array ke copyarray

                bubbleSort(sortArray);
                printf("\n\tSesudah Sorting : "); printArray(sortArray);

                printf("\n\tCari : "); scanf("%d",&find);
                found = sequentialSearch(sortArray, find);

                if(found == -1) printf("\n\tData tidak ditemukan [!]");
                else printf("\n\tData %d ditemukan di indeks ke-%d", sortArray[found], found);
            break;

            case 0:
                printf("\n\tNama pratikan | NPM | Kelas");
                break;

            default:
                printf("\n\tMenu tidak ada!");
                break;
        }getch();

    }while(menu!=0);

    return 0;
}
```

~~ Selamat sudah sampai ke modul akhir, you did well. ~~