

Tugas Jobsheet 11



Dosen pengampu : Randi Proska Sandra, M.Sc

Kode Kelas : 202323430158

Disusun Oleh :

**Fajrul Huda Ash Shiddiq
23343063**

**PROGRAM STUDI INFORMATIKA (NK)
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2023**

Implementasikan data mahasiswa dengan pencarian data menggunakan metode sequential search dan binary search

1. Source Code

```
//Create by_Fajrul Huda Ash Shiddiq_23343063
#include <stdio.h>
#include <string.h>

#define MAX_STUDENTS 100

struct Mahasiswa {
    int NIM;
    char Nama[50];
    char TTL[50];
    float IPK;
};

// Fungsi pencarian sequential search
int sequentialSearch(struct Mahasiswa data[], int n, int key) {
    int i;
    for (i = 0; i < n; i++) {
        if (data[i].NIM == key) {
            return i; // Return index jika data ditemukan
        }
    }
    return -1; // Return -1 jika data tidak ditemukan
}

// Fungsi pencarian binary search
int binarySearch(struct Mahasiswa data[], int low, int high, int key) {
    while (low <= high) {
```

```

    int mid = (low + high) / 2;
    if (data[mid].NIM == key) {
        return mid; // Return index jika data ditemukan
    } else if (data[mid].NIM < key) {
        low = mid + 1;
    } else {
        high = mid - 1;
    }
}

return -1; // Return -1 jika data tidak ditemukan
}

void sortMahasiswa(struct Mahasiswa data[], int n) {
    struct Mahasiswa sorted[MAX_STUDENTS]; // Declare sorted array
    memcpy(sorted, data, n * sizeof(struct Mahasiswa)); // Copy data
    to a temporary array for sorting

    // Sorting based on NIM
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (sorted[j].NIM > sorted[j + 1].NIM) {
                struct Mahasiswa temp = sorted[j];
                sorted[j] = sorted[j + 1];
                sorted[j + 1] = temp;
            }
        }
    }
}

int main() {
    struct Mahasiswa data[MAX_STUDENTS];

```

```

int n, i, key;

printf("Masukkan jumlah data mahasiswa: ");
scanf("%d", &n);

getchar(); // Consume the newline character left in the input
buffer

printf("\nMasukkan data mahasiswa:");
for (i = 0; i < n; i++) {
    printf("\nData ke-%d:\n", i+1);
    printf("NIM: ");
    scanf("%d", &data[i].NIM);
    getchar(); // Consume the newline character left in the
input buffer
    printf("Nama: ");
    fgets(data[i].Nama, sizeof(data[i].Nama), stdin);
    data[i].Nama[strcspn(data[i].Nama, "\n")] = '\0'; // Remove
trailing newline
    printf("TTL: ");
    fgets(data[i].TTL, sizeof(data[i].TTL), stdin);
    data[i].TTL[strcspn(data[i].TTL, "\n")] = '\0'; // Remove
trailing newline
    printf("IPK: ");
    scanf("%f", &data[i].IPK);
}

printf("\nMasukkan NIM yang ingin dicari: ");
scanf("%d", &key);

// Sequential search
int seqResult = sequentialSearch(data, n, key);
if (seqResult != -1) {

```

```

        printf("\nSequential Search: Data ditemukan pada index
%d\n", seqResult);
        printf("NIM: %d\n", data[seqResult].NIM);
        printf("Nama: %s\n", data[seqResult].Nama);
        printf("TTL: %s\n", data[seqResult].TTL);
        printf("IPK: %.2f\n", data[seqResult].IPK);

    } else {
        printf("\nSequential Search: Data tidak ditemukan\n");
    }

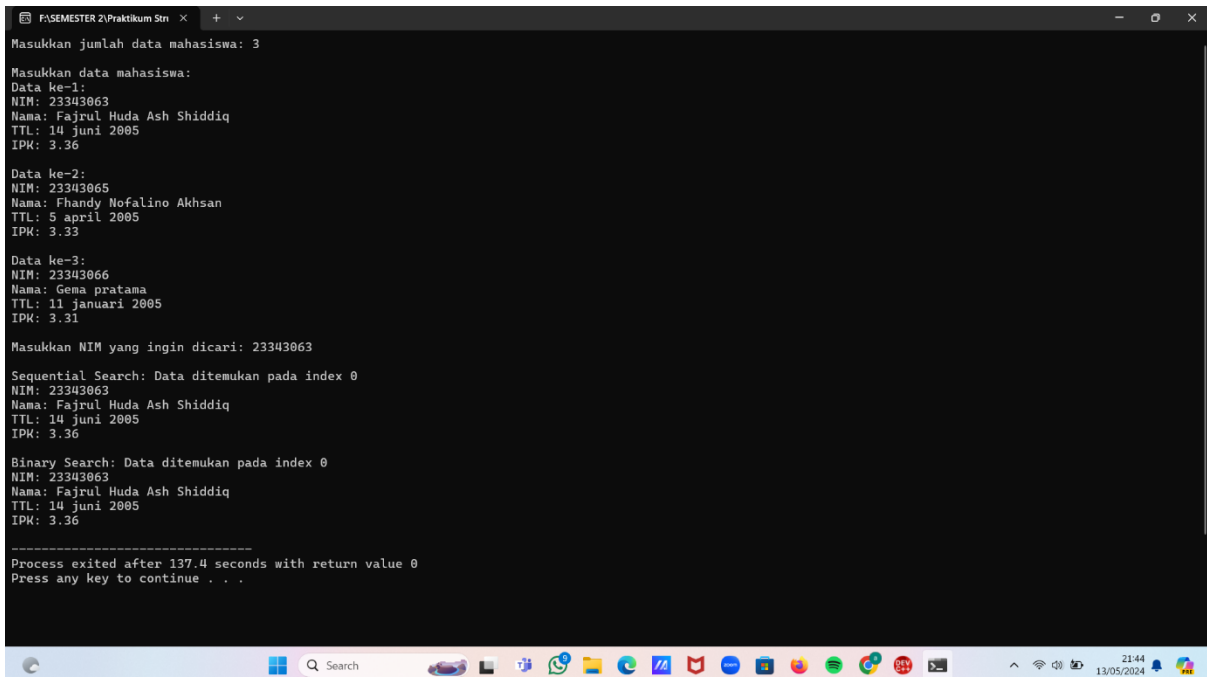
    // Sort the data
    sortMahasiswa(data, n);

    // Binary search
    int binResult = binarySearch(data, 0, n - 1, key);
    if (binResult != -1) {
        printf("\nBinary Search: Data ditemukan pada index %d\n",
binResult);
        printf("NIM: %d\n", data[binResult].NIM);
        printf("Nama: %s\n", data[binResult].Nama);
        printf("TTL: %s\n", data[binResult].TTL);
        printf("IPK: %.2f\n", data[binResult].IPK);
    } else {
        printf("\nBinary Search: Data tidak ditemukan\n");
    }

    return 0;
}

```

2. Screenshot



```
FASEMESTER 2/Praktikum Stn x + v
Masukkan jumlah data mahasiswa: 3
Masukkan data mahasiswa:
Data ke-1:
NIM: 23343063
Nama: Fajrul Huda Ash Shiddiq
TTL: 14 juni 2005
IPK: 3.36
Data ke-2:
NIM: 23343065
Nama: Fhandy Mofalino Akhsan
TTL: 5 april 2005
IPK: 3.33
Data ke-3:
NIM: 23343066
Nama: Gema pratama
TTL: 11 januari 2005
IPK: 3.31
Masukkan NIM yang ingin dicari: 23343063
Sequential Search: Data ditemukan pada index 0
NIM: 23343063
Nama: Fajrul Huda Ash Shiddiq
TTL: 14 juni 2005
IPK: 3.36
Binary Search: Data ditemukan pada index 0
NIM: 23343063
Nama: Fajrul Huda Ash Shiddiq
TTL: 14 juni 2005
IPK: 3.36
-----
Process exited after 137.4 seconds with return value 0
Press any key to continue . . .
```

3. Penjelasan

1. Fungsi “*sequentialSearch*”:

- Fungsi ini melakukan pencarian sekuensial untuk mencari NIM (Nomor Induk Mahasiswa) tertentu dalam array dari struct Mahasiswa.
- Fungsi ini melakukan iterasi melalui array menggunakan sebuah loop, dimulai dari elemen pertama.
- Untuk setiap elemen, fungsi ini membandingkan NIM dengan key menggunakan operator `==`.
- Jika sebuah kecocokan ditemukan, maka fungsi ini akan mengembalikan indeks dari elemen di mana kecocokan terjadi.
- Jika tidak ada kecocokan yang ditemukan setelah melakukan iterasi melalui seluruh array, maka fungsi ini akan mengembalikan -1 yang menunjukkan bahwa kunci tidak ditemukan.

2. Fungsi “*binarySearch*”:

- Fungsi ini melakukan pencarian biner untuk mencari NIM (Nomor Induk Mahasiswa) tertentu dalam sebuah array terurut dari struct Mahasiswa.
- Fungsi ini berulang kali membagi interval pencarian menjadi dua hingga kunci ditemukan atau interval pencarian menjadi kosong.
- Fungsi ini menghitung indeks mid sebagai rata-rata dari low dan high.
- Jika NIM pada indeks mid cocok dengan key, maka fungsi ini akan mengembalikan indeks dari elemen di mana kecocokan terjadi.
- Jika NIM pada indeks mid kurang dari key, maka fungsi ini akan menetapkan `low = mid + 1` untuk mencari di separuh kanan dari array.

- Jika NIM pada indeks mid lebih besar dari key, maka fungsi ini akan menetapkan $high = mid - 1$ untuk mencari di separuh kiri dari array.
- Proses ini akan terus dilakukan hingga kunci ditemukan atau interval pencarian menjadi kosong.
- Jika kunci tidak ditemukan setelah pencarian, maka fungsi ini akan mengembalikan -1 yang menunjukkan bahwa kunci tidak ditemukan.

3. Fungsi sortMahasiswa:

- Fungsi ini mengurutkan array dari struct Mahasiswa berdasarkan NIM mereka menggunakan algoritma bubble sort.
- Fungsi ini membuat sebuah array sementara sorted dan menyalin data dari array asli data ke dalamnya menggunakan memcpy.
- Kemudian, fungsi ini melakukan algoritma bubble sort untuk mengurutkan array sorted berdasarkan NIM dari setiap elemennya.
- Algoritma bubble sort melakukan iterasi melalui array beberapa kali, membandingkan elemen-elemen yang bersebelahan dan menukar mereka jika mereka berada dalam urutan yang salah.
- Setelah pengurutan, array sorted akan berisi data yang diurutkan secara menaik berdasarkan NIM.

4. Cara kerja program

- Program meminta pengguna untuk memasukkan jumlah data mahasiswa yang akan dimasukkan.
- Setelah pengguna memasukkan jumlah data, program meminta pengguna untuk memasukkan data untuk setiap mahasiswa, termasuk NIM, nama, TTL, dan IPK. Untuk memasukkan nama dan TTL yang mungkin mengandung spasi, program menggunakan fgets() untuk membaca input.
- Setelah semua data dimasukkan, program meminta pengguna untuk memasukkan NIM yang ingin dicari.
- Program menggunakan fungsi sequentialSearch untuk mencari NIM yang dimasukkan pengguna secara sekuensial dalam array data mahasiswa. Hasil pencarian ini kemudian ditampilkan ke layar.
- Selanjutnya, program menggunakan fungsi sortMahasiswa untuk mengurutkan array data mahasiswa berdasarkan NIM mereka.
- Setelah data diurutkan, program menggunakan fungsi binarySearch untuk mencari NIM yang dimasukkan pengguna dalam array data mahasiswa yang telah diurutkan. Hasil pencarian ini juga ditampilkan ke layar.
- Program selesai dieksekusi dan mengembalikan nilai 0, menandakan bahwa program telah berjalan dengan sukses.