

PRAKTIKUM STRUKTUR DATA
Tugas Jobsheet 11



Dosen Pengampu :
Randi Proska Sandra, M.Sc

Disusun Oleh :
Joel Wiseda Simanungkalit
23343071

PROGRAM STUDI INFORMATIKA (NK)
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2024

A. Sourcecode

```
// Created by Joel Wiseda Simanungkalit
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Struktur untuk menyimpan data mahasiswa
typedef struct {
    int NIM;
    char nama[50];
    char TTL[50];
    float IPK;
} Mahasiswa;

// Fungsi untuk menambahkan data mahasiswa
void tambahMahasiswa(Mahasiswa mahasiswa[], int* jumlah) {
    printf("Masukkan NIM: ");
    scanf("%d", &mahasiswa[*jumlah].NIM);
    printf("Masukkan Nama: ");
    scanf(" %[^\\n]*c", mahasiswa[*jumlah].nama); // Menggunakan scanf untuk
input string dengan spasi
    printf("Masukkan TTL: ");
    scanf(" %[^\\n]*c", mahasiswa[*jumlah].TTL);
    printf("Masukkan IPK: ");
    scanf("%f", &mahasiswa[*jumlah].IPK);
    (*jumlah)++;
}

// Fungsi untuk menampilkan data mahasiswa
void tampilkanMahasiswa(Mahasiswa mahasiswa[], int jumlah) {
    for (int i = 0; i < jumlah; i++) {
        printf("NIM: %d\\n", mahasiswa[i].NIM);
        printf("Nama: %s\\n", mahasiswa[i].nama);
        printf("TTL: %s\\n", mahasiswa[i].TTL);
        printf("IPK: %.2f\\n\\n", mahasiswa[i].IPK);
    }
}

// Fungsi untuk pencarian sequential berdasarkan NIM
void sequentialSearch(Mahasiswa mahasiswa[], int jumlah, int NIM) {
    for (int i = 0; i < jumlah; i++) {
        if (mahasiswa[i].NIM == NIM) {
            printf("Data ditemukan:\\n");
            printf("NIM: %d\\n", mahasiswa[i].NIM);
            printf("Nama: %s\\n", mahasiswa[i].nama);
            printf("TTL: %s\\n", mahasiswa[i].TTL);
            printf("IPK: %.2f\\n\\n", mahasiswa[i].IPK);
            return;
        }
    }
}
```

```

    }
}
printf("Data tidak ditemukan\n");
}

// Fungsi untuk mengurutkan data berdasarkan NIM (untuk binary search)
void sortMahasiswa(Mahasiswa mahasiswa[], int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = 0; j < jumlah - i - 1; j++) {
            if (mahasiswa[j].NIM > mahasiswa[j + 1].NIM) {
                Mahasiswa temp = mahasiswa[j];
                mahasiswa[j] = mahasiswa[j + 1];
                mahasiswa[j + 1] = temp;
            }
        }
    }
}

// Fungsi untuk pencarian binary berdasarkan NIM
void binarySearch(Mahasiswa mahasiswa[], int jumlah, int NIM) {
    int left = 0;
    int right = jumlah - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (mahasiswa[mid].NIM == NIM) {
            printf("Data ditemukan:\n");
            printf("NIM: %d\n", mahasiswa[mid].NIM);
            printf("Nama: %s\n", mahasiswa[mid].nama);
            printf("TTL: %s\n", mahasiswa[mid].TTL);
            printf("IPK: %.2f\n\n", mahasiswa[mid].IPK);
            return;
        } else if (mahasiswa[mid].NIM < NIM) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }
    printf("Data tidak ditemukan\n");
}

int main() {
    Mahasiswa mahasiswa[100];
    int jumlah = 0;
    int pilihan;
    int NIM;

    do {
        printf("Menu:\n");

```

```

printf("1. Tambah Mahasiswa\n");
printf("2. Tampilkan Mahasiswa\n");
printf("3. Pencarian Sequential\n");
printf("4. Pencarian Binary\n");
printf("5. Keluar\n");
printf("Pilih: ");
scanf("%d", &pilihan);

switch (pilihan) {
    case 1:
        tambahMahasiswa(mahasiswa, &jumlah);
        break;
    case 2:
        tampilkanMahasiswa(mahasiswa, jumlah);
        break;
    case 3:
        printf("Masukkan NIM yang dicari: ");
        scanf("%d", &NIM);
        sequentialSearch(mahasiswa, jumlah, NIM);
        break;
    case 4:
        sortMahasiswa(mahasiswa, jumlah);
        printf("Masukkan NIM yang dicari: ");
        scanf("%d", &NIM);
        binarySearch(mahasiswa, jumlah, NIM);
        break;
    case 5:
        printf("Keluar dari program\n");
        break;
    default:
        printf("Pilihan tidak valid\n");
        break;
}
} while (pilihan != 5);

return 0;
}

```

B. Penjelasan Program

Struktur Data Mahasiswa

Program ini menggunakan struktur data `Mahasiswa` untuk menyimpan informasi tentang mahasiswa, yang meliputi NIM, Nama, TTL, dan IPK.

Fungsi `tambahMahasiswa`

Fungsi ini digunakan untuk menambahkan data mahasiswa baru ke dalam array. Data yang dimasukkan oleh pengguna akan disimpan ke dalam array `mahasiswa`.

Fungsi `tampilkanMahasiswa`

Fungsi ini digunakan untuk menampilkan semua data mahasiswa yang ada dalam array. Setiap data mahasiswa akan dicetak ke layar.

Fungsi `sequentialSearch`

Fungsi ini melakukan pencarian data mahasiswa berdasarkan NIM menggunakan metode sequential search. Fungsi ini akan mencari data mahasiswa satu per satu hingga ditemukan atau mencapai akhir array.

Fungsi `sortMahasiswa`

Fungsi ini digunakan untuk mengurutkan data mahasiswa berdasarkan NIM menggunakan Bubble Sort. Pengurutan ini diperlukan sebelum melakukan pencarian dengan metode binary search.

Fungsi `binarySearch`

Fungsi ini melakukan pencarian data mahasiswa berdasarkan NIM menggunakan metode binary search. Fungsi ini bekerja dengan membagi array menjadi dua bagian dan mencari NIM di bagian yang tepat hingga data ditemukan atau tidak ditemukan.

C. Output Program

1. Sequential Search

```
D:\sem 2\Prak SD\jobsheet 11  X  +  v
Menu:
1. Tambah Mahasiswa
2. Tampilkan Mahasiswa
3. Pencarian Sequential
4. Pencarian Binary
5. Keluar
Pilih: 2
NIM: 23343071
Nama: Joel
TTL: 04-04-2005
IPK: 3.90

NIM: 23343065
Nama: wanda
TTL: 05-07-2005
IPK: 3.70

NIM: 23343072
Nama: Regina
TTL: 02-04-2005
IPK: 3.80

Menu:
1. Tambah Mahasiswa
2. Tampilkan Mahasiswa
3. Pencarian Sequential
4. Pencarian Binary
5. Keluar
Pilih: 3
Masukkan NIM yang dicari: 23343071
Data ditemukan:
NIM: 23343071
Nama: Joel
TTL: 04-04-2005
IPK: 3.90
```

2. Binary Search

```
D:\sem 2\Prak SD\Jobsheet 11 X + v
Nama: Joel
TTL: 04-04-2005
IPK: 3.90

Menu:
1. Tambah Mahasiswa
2. Tampilkan Mahasiswa
3. Pencarian Sequential
4. Pencarian Binary
5. Keluar
Pilih: 4
Masukkan NIM yang dicari: 23343065
Data ditemukan:
NIM: 23343065
Nama: wanda
TTL: 05-07-2005
IPK: 3.70

Menu:
1. Tambah Mahasiswa
2. Tampilkan Mahasiswa
3. Pencarian Sequential
4. Pencarian Binary
5. Keluar
Pilih: 5
Keluar dari program

-----
Process exited after 617 seconds with return value 0
Press any key to continue . . .
```

Kesimpulan

Implementasi ini memungkinkan pengguna untuk menambahkan data mahasiswa, menampilkan data mahasiswa, dan melakukan pencarian data mahasiswa berdasarkan NIM menggunakan metode sequential search dan binary search. Binary search memerlukan data yang sudah diurutkan, sehingga diperlukan langkah pengurutan sebelum melakukan pencarian. Sequential search tidak memerlukan data yang diurutkan dan melakukan pencarian dengan memeriksa setiap elemen satu per satu.