

Nama : Yanuar Nurul Hilal
NIM : 222112418
Kelas : 2KS4

Penugasan Struktur Data Praktikum 11

1. Modifikasilah file modul11a1.c dan file modul11a2.c supaya dapat mengakomodasi pencarian sekuensial pada data yang terurut, baik urut menaik atau menurun.

```
// modul11a1
#include <stdio.h>
#include <stdbool.h>
#define MAX 100 // ukuran maksimum array

void fill_data(int data[], int *size)
{ // mengisi data
    printf("Input ukuran array (max 100): ");
    scanf("%d", size);
    printf("Input data: ");
    for (int i = 0; i < *size; i++)
    {
        scanf("%d", &data[i]);
    }
}

int seq_search(int nim[], int size, int x)
{
    for (int i = 0; i < size; i++)
    {
        if (nim[i] == x)
            return i;
    }
    return -1;
}

bool cekascending(int data[], int size, int x)
{
    if (size==0 || size==1)
        return true;
    for(int i=1;i<size;i++)
        //unsorted found
        if (data[i-1]>data[i])
            return false;
}
```

```

        // no unsorted found
        return true;
    }

bool cekdescending(int data[],int size)
{
    if (size==0 || size==1)
        return true;
    for(int i=1;i<size;i++)
        //unsorted found
        if (data[i-1]<data[i])
            return false;
    // no unsorted found
    return true;
}

int orderedsearch(int data[],int size,int x)
{
    int i=0;
    if (cekascending(data,size,x))
    {
        while(i<size && data[i]<=x)
        {
            if (data[i]==x)
                return i;
            i++;
        }
        return -1;
    }
    else if (cekdescending(data,size))
    {
        while(i<size && data[i]>=x)
        {
            if (data[i]==x)
                return i;
            i++;
        }
        return -1;
    }
    else
    {

```

```

        return -1;
    }
}

void main()
{
    int data[MAX];
    int size; // ukuran array
    int x;
    fill_data(data, &size);
    printf("Data yang ingin dicari: ");
    scanf("%d", &x);
    if (orderedsearch(data,size, x) == -1) printf("Data tidak ditemukan");
    else printf("Data ditemukan pada indeks
ke-%d",orderedsearch(data,size,x));
}

```

```

//modul11a2
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int value;
    struct node *next;
};

typedef struct node *ptrnode;

ptrnode head = NULL;
int jumnode; // jumlah node

ptrnode insert(int nilai)
{
    ptrnode p, q;
    p = (ptrnode)malloc(sizeof(struct node));
    p->value = nilai;
    p->next = NULL;
    if (head == NULL)
    {
        head = p;
    }
}

```

```

        q = head;
    }
    else
    {
        q = head;
        while (q->next != NULL)
        {
            q = q->next;
        }
        q->next = p;
    }
    return (head);
}

void isi_data()
{
    int k;
    printf("input jumlah node: ");
    scanf("%d", &jumnode);
    for (int j = 1; j <= jumnode; j++)
    {
        printf("input data ke-%d :", j);
        scanf("%d", &k);
        head = insert(k);
    }
}

int search(int x, int metode)
{ // x adalah nilai yang dicari
    int j = 1;
    ptrnode tmp = head;
    while (tmp != NULL)
    {
        if (x == tmp->value)
        {
            return j;
        }
        else if (metode == 1 && x < tmp->value)
        {
            return -1;
        }
        else if (metode == 2 && x > tmp->value)
        {

```

```

        return -1;
    }
    else
    {
        tmp = tmp->next;
        j++;
    }
}
return -1; // jika tidak ada yang dicari return -1
}

```

```

void bersihkan_memori()
{
    while (head != NULL)
    {
        ptrnode tmp = head;
        head = head->next;
        tmp->next = NULL;
        free(tmp);
    }
}

```

```

int metode()
{
    printf("Metode:\n 1.Ascending\n 2.Descending\n");
    int metode;
    printf("Pilih: ");
    scanf("%d", &metode);
    if (metode == 1)
    {
        printf("Metode 1 Terpilih\n\n");
        return 1;
    }
    else
    {
        printf("Metode 2 Terpilih\n\n");
        return 2;
    }
}

```

```

void main()
{
    int metod = metode();
}

```

```

isi_data();
int x;
printf("input nilai yang mau dicari: ");
scanf("%d", &x);

if (search(x,metod) == -1)
    printf("data tidak ditemukan");
else
    printf("data ditemukan di node ke-%d", search(x,metod));
bersihkan_memori();
}

```

2. Buat program untuk pencarian data students berisi int nim, char nama[50] dengan struktur array. pencarian bisa dilakukan secara sekuensial/biner dengan berdasarkan nim atau berdasarkan nama.

```

//modul11a1_no2
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100 // ukuran maksimum array

void fill_nim(int nim[], char nama[][50], int *size)
{ // mengisi nim
    printf("Input ukuran array (max 100): ");
    scanf("%d", size);
    for (int i = 0; i < *size; i++)
    {
        printf("\nMahasiswa ke-[%d] \nNim: ", i+1);
        scanf("%d", &nim[i]);
        getchar();
        printf("Nama: ");
        gets(nama[i]);
    }
}

int nim_search(int nim[], int size, int x)
{
    for (int i = 0; i < size; i++)
    {
        if (nim[i] == x)

```

```

        return i;
    }
    return -1;
}

int nama_search(char nama[][50], int size, char y[])
{
    for (int i = 0; i < size; i++)
    {
        if(strcmp(nama[i],y)==0)
            return i;
    }
    return -1;
}

int metode()
{
    printf("\nPilih Metode Pencarian :\n 1.Nim\n 2.Nama\n");
    int metode;
    printf("\nPilihan Anda : ");
    scanf("%d", &metode);
    if (metode == 1)
    {
        printf("-Metode 1 Terpilih-\n\n");
        return 1;
    }
    else
    {
        printf("-Metode 2 Terpilih-\n\n");
        return 2;
    }
}

void main()
{
    int nim[MAX];
    char nama[MAX][50];
    int size; // ukuran array
    int x;
    char y[50];
    fill_nim(nim, nama,&size);

```

```

    int metod = metode();

    if (metod == 1)
    {
        printf("Nilai yang ingin dicari: ");
        scanf("%d", &x);
        if (nim_search(nim, size, x) == -1)
            printf("Nilai tidak ditemukan");
        else
            printf("Nilai ditemukan pada indeks ke-%d", nim_search(nim,
size, x));
    }
    else if (metod == 2)
    {
        printf("Nama yang ingin dicari: ");
        getchar();
        gets(y);
        if (nama_search(nama, size, y) == -1)
            printf("Nama tidak ditemukan");
        else
            printf("Nama ditemukan pada indeks ke-%d", nama_search(nama,
size, y));
    }
}

```

3. Buat program untuk pencarian data students berisi int nim, char nama[50] dengan struktur linked list. pencarian bisa dilakukan secara sekuensial/biner dengan berdasarkan nim atau berdasarkan nama.

```

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>

struct node
{
    int nim;
    char name[50];
    struct node *next;
};

```



```

typedef struct node *ptrnode;

ptrnode head = NULL;
int countnode; //count of node

ptrnode insert(int nim, char name[])
{
    ptrnode p,q;
    p = (ptrnode)malloc(sizeof(struct node));
    strcpy(p->name, name);
    p->nim = nim;
    p->next = NULL;
    if (head == NULL)
    {
        head = p;
        q = head;
    }
    else
    {
        q = head;
        while (q->next != NULL)
        {
            q = q->next;
        }
        q->next = p;
    }
    return (head);
}

void fillData()
{
    int k;
    char l[50];
    printf("Masukkan jumlah mahasiswa = ");
    scanf("%d", &countnode);
    for (int j = 1; j <= countnode; j++)
    {
        printf("Mahasiswa ke-[%d]\n", j);
        printf("NIM      = ");
        scanf("%d", &k);
        printf("Nama      = ");
        scanf("%s", &l);
        head = insert(k, l);
    }
}

```

```

    }
}

bool checkAscending(ptrnode head)
{
    //function to check linked list is
    //sorted in descending order or not
    if (head == NULL)
        return true;
    //traverse the list till last node and return
    //false if a node is smaller than or equal
    //its next
    ptrnode tmp;
    for (tmp = head; tmp->next != NULL; tmp = tmp->next)
        if (tmp->nim >= tmp->next->nim)
            return false;

    return true;
}

bool checkDescending(ptrnode head)
{
    //function to check linked list is
    //sorted in descending order or not
    if (head == NULL)
        return true;
    //traverse the list till last node and return
    //false if a node is smaller than or equal
    //its next
    ptrnode tmp;
    for (tmp = head; tmp->next != NULL; tmp=tmp->next)
        if (tmp->nim <= tmp->next->nim)
            return false;

    return true;
}

bool ordered_search_name(ptrnode head, char name)
{
    ptrnode current = head; //initialize current
    while (current != NULL)
    {
        if (strcmp(name, head->name) == 0)

```

```

        return true;
        current = current->next;
    }
    return false;
}

int search_name(char name[])
{
    int j = 1;
    ptrnode tmp = head;
    while (tmp != NULL)
    {
        if(strcmp(name, tmp->name) == 0)
        {
            return j;
        }
        else
        {
            tmp = tmp->next;
            j++;
        }
    }
    return -1; //if the value we are looking for doesn't exist, return -1
}

int ordered_search_nim(int x)
{
    int i = 1;
    ptrnode tmp = head;

    if (checkAscending(head))
    {
        while (tmp != NULL && i < &countnode && tmp->nim <= x)
        {
            if(x == tmp->nim)
            {
                return i;
            }
            else
            {
                tmp = tmp->next;
                i++;
            }
        }
    }
}

```

```

    }
}
else if (checkDescending(head))
{
    while (tmp != NULL && i < &countnode && tmp->nim >= x)
    {
        if(x == tmp->nim)
        {
            return i;
        }
        else
        {
            tmp = tmp->next;
            i++;
        }
    }
}
else
{
    int j = 1;
    ptrnode tmp = head;
    while (tmp != NULL)
    {
        if(x == tmp->nim)
        {
            return j;
        }
        else
        {
            tmp = tmp->next;
            j++;
        }
    }
    return -1; //if the value we are looking for doesn't exist, return
-1
}
}

void freeMemory()
{
    while (head != NULL)
    {
        ptrnode tmp = head;

```

```

        head = head->next;
        tmp->next = NULL;
        free(tmp);
    }
}

void display(ptrnode head) //display the data in linked list
{
    ptrnode tmp = head;
    int i=1;
    while (tmp != NULL)
    {
        printf("\nMahasiswa ke-[%d]\n", i);
        printf("Nama   = %s\n", tmp->name);
        printf("NIM    = %d", tmp->nim);
        tmp = tmp->next;
        i++;
    }
}

void main()
{
    char x[50];
    int choice, c, y;
    //Make a menu
    do
    {
        printf("Sequential search pada Linked list\n");
        printf("1. Insert Data\n");
        printf("2. Pencarian berdasarkan Nama\n");
        printf("3. Pencarian berdasarkan NIM\n");
        printf("4. Display daftar mahasiswa\n");
        printf("Pilihan Anda? = ");
        scanf("%d", &choice);

        switch(choice)
        {
            printf(" ");
            case 1 :
                fillData();
                break;
            case 2 :
                printf("Masukkan nama yang ingin dicari = ");

```

```

        scanf("%s", &x);
        if (search_name(x) == -1)
            printf("Mahasiswa dengan nama (%s) tidak ditemukan",
x);
        else
            printf("Mahasiswa dengan nama (%s) ada pada node
ke-[%d]", x, search_name(x));
            break;
        case 3 :
            printf("Masukkan nim yang ingin dicari = ");
            scanf("%d", &y);
            if (ordered_search_nim(y) == -1)
                printf("Mahasiswa dengan NIM (%d) tidak ditemukan", y);
            else
                printf("Mahasiswa dengan NIM (%d) ada pada node
ke-[%d]", y, ordered_search_nim(y));
                break;
        case 4 :
            display(head);
            break;
        default :
            printf("Pilihan tidak valid\n");
            break;
    }
    printf("\n Apakah ingin melanjutkan? (tekan 1 untuk 'YA') = ");
    scanf("%d", &c);
} while (c == 1);
getchar();
return 0;

freeMemory();
}

```