

IDENTITAS DIRI

Nama : Fadhil Rahmat
Nim : 22343045
Mata kuliah : Pratikum Struktur Data
Sesi : 202223430030

Nama program

Penyimpanan data nama nama murid serta dapat menghapus data yang telah disimpan

Latar belakang

Program ini dapat menambahkan data murid, menampilkan data murid, menghapus data murid, mencari data murid, mengurutkan data murid dan menghapus seluruh data murid.

```

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

struct Murid {
    char nama[50];
    struct Murid* next;
};

struct Murid* buatSimpul(char nama[]) {
    struct Murid* simpulBaru = (struct Murid*)malloc(sizeof(struct Murid));
    strcpy(simpulBaru->nama, nama);
    simpulBaru->next = NULL;
    return simpulBaru;
}

void tambahData(struct Murid** head, char nama[]) {
    struct Murid* simpulBaru = buatSimpul(nama);
    if (*head == NULL) {
        *head = simpulBaru;
    } else {
        struct Murid* saat_ini = *head;
        while (saat_ini->next != NULL) {
            saat_ini = saat_ini->next;
        }
        saat_ini->next = simpulBaru;
    }
    printf("Data Murid %s telah ditambahkan.\n", nama);
}

void tampilkanData(const struct Murid* head) {
    if (head == NULL) {
        printf("Linked list kosong.\n");
        return;
    }
    printf("Data Murid:\n");
    printf("Nama\n");
    const struct Murid* saat_ini = head;
    while (saat_ini != NULL) {
        printf("%s\n", saat_ini->nama);
        saat_ini = saat_ini->next;
    }
}

```

```
    }  
}
```

```
void hapusData(struct Murid** head, char nama[]) {  
    if (*head == NULL) {  
        printf("Linked list kosong.\n");  
        return;  
    }  
    struct Murid* saat_ini = *head;  
    struct Murid* previous = NULL;  
    while (saat_ini != NULL) {  
        if (strcmp(saat_ini->nama, nama) == 0) {  
            if (previous == NULL) {  
                *head = saat_ini->next;  
            } else {  
                previous->next = saat_ini->next;  
            }  
            free(saat_ini);  
            printf("Data Murid %s telah dihapus.\n", nama);  
            return;  
        }  
        previous = saat_ini;  
        saat_ini = saat_ini->next;  
    }  
    printf("Data Murid %s tidak ditemukan.\n", nama);  
}
```

```
void cariData(const struct Murid* head, char nama[]) {  
    if (head == NULL) {  
        printf("Linked list kosong.\n");  
        return;  
    }  
    const struct Murid* saat_ini = head;  
    int found = 0;  
    while (saat_ini != NULL) {  
        if (strcmp(saat_ini->nama, nama) == 0) {  
            printf("Data Murid %s ditemukan.\n", nama);  
            found = 1;  
        }  
        saat_ini = saat_ini->next;  
    }  
}
```

```

    if (!found) {
        printf("Data Murid %s tidak ditemukan.\n", nama);
    }
}

void urutkanData(struct Murid** head) {
    if (*head == NULL) {
        printf("Linked list kosong.\n");
        return;
    }
    int swapped;
    struct Murid* saat_ini;
    struct Murid* last = NULL;
    do {
        swapped = 0;
        saat_ini = *head;
        while (saat_ini->next != last) {
            if (strcmp(saat_ini->nama, saat_ini->next->nama) > 0) {
                char temp[50];
                strcpy(temp, saat_ini->nama);
                strcpy(saat_ini->nama, saat_ini->next->nama);
                strcpy(saat_ini->next->nama, temp);
                swapped = 1;
            }
            saat_ini = saat_ini->next;
        }
        last = saat_ini;
    } while (swapped);
    printf("Linked list telah diurutkan berdasarkan nama Murid.\n");
}

```

```

void ubahData(struct Murid* head, char nama[]) {
    if (head == NULL) {
        printf("Linked list kosong.\n");
        return;
    }
    struct Murid* saat_ini = head;
    int found = 0;
    while (saat_ini != NULL) {
        if (strcmp(saat_ini->nama, nama) == 0) {
            printf("Data Murid %s ditemukan.\n", nama);
        }
        saat_ini = saat_ini->next;
    }
}

```

```

        printf("Masukkan data yang baru:\n");
        printf("Nama: ");
        scanf("%s", saat_ini->nama);
        printf("Data Murid %s telah diubah.\n", nama);
        found = 1;
    }
    saat_ini = saat_ini->next;
}
if (!found) {
    printf("Data Murid %s tidak ditemukan.\n", nama);
}
}

```

```

void hapusSeluruhData(struct Murid** head) {
    struct Murid* saat_ini = *head;
    struct Murid* next;
    while (saat_ini != NULL) {
        next = saat_ini->next;
        free(saat_ini);
        saat_ini = next;
    }
    *head = NULL;
    printf("Seluruh data Murid telah dihapus.\n");
}

```

```

int main() {
    struct Murid* head = NULL;
    int choice;
    char nama[50];
    printf("Nama          : Fadhil Rahmat\n");
    printf("NIM           : 2234045 \n");
    printf("Mata Kuliah      : Pratikum Struktur Data\n");
    while (1) {
        printf("\nMenu:\n");
        printf("=====\n");
        printf("1. Tambahkan Data Murid\n");
        printf("2. Tampilkan Data Murid\n");
        printf("3. Hapus Data Murid\n");
        printf("4. Cari Data Murid\n");
        printf("5. Urutkan Data Murid\n");
        printf("6. Ubah Data Murid\n");
    }
}

```

```

printf("7. Hapus Seluruh Data Murid\n");
printf("0. Keluar\n");
printf("=====\n");
printf("Masukkan pilihan Anda: ");
scanf("%d", &choice);

switch (choice) {
    case 0:
        printf("Terima kasih!\n");
        exit(0);
    case 1:
        printf("Masukkan nama Murid: ");
        scanf("%s", nama);
        tambahData(&head, nama);
        break;
    case 2:
        tampilkanData(head);
        break;
    case 3:
        printf("Masukkan nama Murid yang ingin dihapus: ");
        scanf("%s", nama);
        hapusData(&head, nama);
        break;
    case 4:
        printf("Masukkan nama Murid yang ingin dicari: ");
        scanf("%s", nama);
        cariData(head, nama);
        break;
    case 5:
        urutkanData(&head);
        break;
    case 6:
        printf("Masukkan nama Murid yang ingin diubah: ");
        scanf("%s", nama);
        ubahData(head, nama);
        break;
    case 7:
        hapusSeluruhData(&head);
        break;
    default:
        printf("Pilihan tidak valid.\n");

```

```

    }
}

return 0;
}

```

