

29 Februari 2024

PRAKTIKUM 2.2

Mata Kuliah Algoritma dan Struktur Data



Nama Dosen Pengampu

Ibu Umi Sa'adah S.Kom, M.Kom

Dikerjakan oleh

Nama : Muhammad Rafi Rizaldi

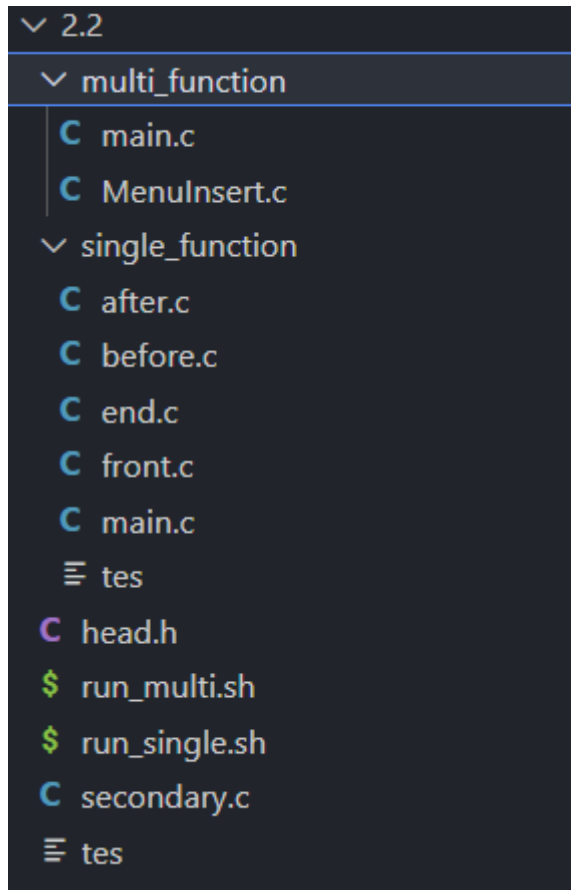
NRP : 3123600001

Kelas : 1 D4 IT A

Untuk source code menurut saya lebih jelas kalau langsung lihat di github saya

[link disini](#)

PERCOBAAN :



Saya membuat folder 2.2 untuk praktikum kali ini, di dalam folder ini juga berisi 2 folder lagi, yaitu single function (untuk no 1 - 4) dan multi function (untuk no 5). File file dalam folder 2.2 adalah sebagai berikut :

- **head.h** (berfungsi sebagai header yang akan meng-include semua file agar fungsi fungsi yang saya buat bisa diakses antar file serta mendeklarasikan semua variabel dan prototype fungsi yang akan digunakan)

```
#ifndef HEAD_H
#define HEAD_H

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

```

struct Node
{
    int nomor;
    char nama[50];
    int nilai;
    struct Node *next;
};

typedef struct Node Siswa;

extern Siswa *head;
extern Siswa *current;

extern int failedInBefore;
extern int failedInAfter;

void clearScreen();
void alokasi();
void tampil();
void free_memory();
void awal();
void akhir();
void after();
void before();
void MenuInsert();

#endif

```

- **secondary.c** (berisikan fungsi fungsi sampingan yang akan dipakai dalam linked list, seperti alokasi, tampil, dan free_memory)

```

#include "head.h"

Siswa *head = NULL, *current;
int failedInAfter;
int failedInBefore;

void alokasi()
{
    current = (Siswa *)malloc(sizeof(Siswa));
    if (current == NULL)
    {
        printf("gagal mengalokasikan memori\n");
        exit(1);
    }
}

```

```

    }

    printf("\nNo\t: ");
    scanf("%d", &current->nomor);
    getchar();
    printf("Nama\t: ");
    fgets(current->nama, sizeof(current->nama), stdin);
    current->nama[strcspn(current->nama, "\n")] = 0;
    printf("Nilai\t: ");
    scanf("%d", &current->nilai);
    getchar();
    current->next = NULL;
}

void tampil()
{
    Siswa *baca;
    baca = head;
    printf("\nData dalam SLL :\n");
    while (baca != NULL)
    {
        printf("\nNo\t= %d\n", baca->nomor);
        printf("Nama\t= %s\n", baca->nama);
        printf("Nilai\t= %d\n", baca->nilai);
        baca = baca->next;
    }
    printf("\n");
}

void free_memory()
{
    Siswa *temp;
    while (head != NULL)
    {
        temp = head;
        head = head->next;
        free(temp);
    }
}

void clearScreen()
{
#ifdef _WIN32
    system("cls"); // For Windows

```

```

#else
    printf("\033[2J\033[H"); // For Linux/Unix
#endif
}

```

Jadi saya hanya punya satu file yang berisi fungsi main untuk single function (no 1 - 4) dan satu lagi untuk menu insert / multi function (no 5)

file : single_function/main.c (saya buat menjadi switch case untuk pemilihan insert agar tidak terjadi pengulangan dalam penulisan kode saya)

```

#include "../head.h"

int main(int argc, char const *argv[])
{
    char choice;
    int insert_choice;
    printf("SINGLE LINKED LIST INSERT\n");
    printf("SINGLE FUNCTION MODE : \n1.
FRONT\n2.END\n3.AFTER\n4.BEFORE\n");
    printf("INPUT YOUR CHOICE : ");
    scanf("%d", &insert_choice);

    switch (insert_choice)
    {
        case 1:
            printf("Linked list untuk aplikasi INSERT DI AWAL\n");
            do
            {
                awal();
                printf("Lagi? (y/t) : ");
                choice = getchar();
            } while (choice == 'y' || choice == 'Y');
            break;
        case 2:
            printf("Linked list untuk aplikasi INSERT DI AKHIR\n");
            do
            {
                akhir();
                printf("Lagi? (y/t) : ");
                choice = getchar();
            } while (choice == 'y' || choice == 'Y');
            break;
    }
}

```

```

        } while (choice == 'y' || choice == 'Y');
        break;
    case 3:
        printf("Linked list untuk aplikasi INSERT SETELAH NILAI
TERTENTU, INSERT AKHIR TERLEBIH DAHULU\n");
        do
        {
            akhir();
            printf("Lagi? (y/t) : ");
            choice = getchar();
        } while (choice == 'y' || choice == 'Y');
        printf("INSERT AFTER\n");
        after();
        break;
    case 4:
        printf("Linked list untuk aplikasi INSERT SEBELUM NILAI
TERTENTU, INSERT AKHIR TERLEBIH DAHULU\n");
        do
        {
            akhir();
            printf("Lagi? (y/t) : ");
            choice = getchar();
        } while (choice == 'y' || choice == 'Y');
        printf("INSERT BEFORE\n");
        before();
        break;
    default:
        break;
    }
    clearScreen();
    tampil();
    free_memory();
    return 0;
}

```

Lalu isi dari fungsi fungsi lainnya akan berisi seperti nama file nya, seperti berikut :

a. INSERT AWAL

file : **front.c**

```
#include "../head.h"
```

```

void awal()
{
    alokasi();
    if (head != NULL)
    {
        current->next = head;
    }
    head = current;
    clearScreen();
}

```

Input :

```

Linked list untuk aplikasi INSERT DI AWAL

```

```

No      : 1
Nama    : aldi
Nilai   : 98
Lagi? (y/t) : y

```

```

No      : 2
Nama    : lola
Nilai   : 77
Lagi? (y/t) : y

```

```

No      : 3
Nama    : kiko
Nilai   : 88
Lagi? (y/t) : 

```

Output :

```

Data dalam SLL :

```

```

No      = 3
Nama    = kiko
Nilai   = 88

```

```

No      = 2
Nama    = lola
Nilai   = 77

```

```

No      = 1
Nama    = aldi
Nilai   = 98

```

b. INSERT AKHIR

file : end.c

```

#include "../head.h"

void akhir()
{
    Siswa *tail;
    alokasi();
    if (head == NULL)
        head = current;
    else
    {
        tail = head;
        while (tail->next != NULL)
            tail = tail->next;
        tail->next = current;
    }
    clearScreen();
}

```

Input :

Linked list untuk aplikasi INSERT DI AKHIR

No : 1
 Nama : aldi
 Nilai : 88
 Lagi? (y/t) : y

No : 2
 Nama : lola
 Nilai : 78
 Lagi? (y/t) : y

No : 3
 Nama : kiko
 Nilai : 99
 Lagi? (y/t) : t

Output :

Data dalam SLL :

No = 1
Nama = aldi
Nilai = 88

No = 2
Nama = lola
Nilai = 78

No = 3
Nama = kiko
Nilai = 99

c. INSERT AFTER

file : after.c

```
#include "../head.h"

void after()
{
    Siswa *findKey;
    int key;

    alokasi();
    findKey = head;
    printf("Masukkan setelah nilai ? ");
    scanf("%d", &key);
    while (findKey != NULL && findKey->nomor != key)
    {
        findKey = findKey->next;
    }
    if (findKey == NULL)
    {
        printf("Gagal menemukan key\n");
        failedInAfter = 1;
    }
    else
    {
        current->next = findKey->next;
        findKey->next = current;
    }
    clearScreen();
}
```

Input :

```
Linked list untuk aplikasi INSERT SETELAH NILAI TERTENTU, INSERT AKHIR TERLEBIH DAHULU

No      : 1
Nama    : ALDI
Nilai   : 89
Lagi? (y/t) : Y

No      : 2
Nama    : KIKO
Nilai   : 99
Lagi? (y/t) : Y

No      : 3
Nama    : LOLA
Nilai   : 67
Lagi? (y/t) : T
INSERT AFTER

No      : 12
Nama    : PAPA
Nilai   : 88
Masukkan setelah nilai ? 2
```

Output :

Data dalam SLL :

No = 1
Nama = ALDI
Nilai = 89

No = 2
Nama = KIKO
Nilai = 99

No = 12
Nama = PAPA
Nilai = 88

No = 3
Nama = LOLA
Nilai = 67

d. INSERT BEFORE

file : before.c

```
#include "../head.h"
// comment out the unused part
void before()
{
    Siswa *findKey, *prev;

    prev = NULL;
```

```

    findKey = head;

    int key;
    alokasi();
    printf("Masukkan sebelum nilai ? ");
    scanf("%d", &key);

    while (findKey != NULL && findKey->nomor != key)
    {
        prev = findKey;
        findKey = findKey->next;
    }
    if (findKey == NULL) {
        printf("Gagal menemukan key\n");
        failedInBefore = 1;
    }
    else
    {
        if (prev == NULL)
        {
            current->next = head;
            head = current;
        }
        else
        {
            current->next = prev->next;
            prev->next = current;
        }
    }
    clearScreen();
}

```

Input :

```
Linked list untuk aplikasi INSERT SEBELUM NILAI TERTENTU, INSERT AKHIR TERLEBIH DAHULU
```

```
No      : 1  
Nama    : aldi  
Nilai   : 87  
Lagi? (y/t) : y
```

```
No      : 2  
Nama    : kiko  
Nilai   : 99  
Lagi? (y/t) : y
```

```
No      : 4  
Nama    : lola  
Nilai   : 77  
Lagi? (y/t) : t  
INSERT BEFORE
```

```
No      : 3  
Nama    : hari  
Nilai   : 66  
Masukkan sebelum nilai ? 4
```

Output :

```
Data dalam SLL :
```

```
No      = 1  
Nama    = aldi  
Nilai   = 87
```

```
No      = 2  
Nama    = kiko  
Nilai   = 99
```

```
No      = 3  
Nama    = hari  
Nilai   = 66
```

```
No      = 4  
Nama    = lola  
Nilai   = 77
```

Kemudian untuk nomor 5 (Menu Insert) terdapat dalam folder multi_function. Di dalamnya hanya berisi file main.c dan menuinsert.c, karena nanti fungsi awal, akhir, after, before nya akan mengambil dari file file di dalam folder single_function

- file : MenuInsert.c

```
#include "../head.h"  
  
void MenuInsert()  
{  
    int exit = 0;  
    int insert_choice;  
    while (!exit)  
    {
```

```

        failedInAfter = 0;
        failedInBefore = 0;
        printf("1. AWAL\n2. AKHIR\n3. AFTER\n4. BEFORE\n0. exit\n");
        printf("Masukkan pilihan : ");
        scanf("%d", &insert_choice);
        switch (insert_choice)
        {
        case 1:
            awal();
            break;
        case 2:
            akhir();
            break;
        case 3:
            (head != NULL) ? after() : (clearScreen(), printf("SLL
Masih kosong, tidak bisa insert after\n"));
            printf(failedInAfter == 1 ? "key tidak ditemukan\n" : "");
            break;
        case 4:
            (head != NULL) ? before() : (clearScreen(), printf("SLL
Masih kosong, tidak bisa insert before\n"));
            printf(failedInBefore == 1 ? "key tidak ditemukan\n" : "");
            break;
        case 0:
            exit = 1;
            break;
        default:
            clearScreen();
            printf("Pilihan anda tadi tidak valid, silahkan coba
lagi\n");
            break;
        }
        tampil();
    }
}

```

- file : multi_function/main.c

```

#include "../head.h"

int main(int argc, char const *argv[])
{
    MenuInsert();
}

```

```

    free_memory();
    return 0;
}

```

Output :

```

○ mrrizzza@Dravarezld:/mnt/d/KULIAH2-ASD/ASD/SLL/2.2$
1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit
Masukkan pilihan : 1

No      : 1
Nama    : zai
Nilai   : 89

Data dalam SLL :

No      = 1
Nama    = zai
Nilai   = 89

1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit
Masukkan pilihan : 2

No      : 4
Nama    : dan
Nilai   : 68

Data dalam SLL :

No      = 1
Nama    = zai
Nilai   = 89

No      = 4
Nama    = dan
Nilai   = 68

```

```

1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit
Masukkan pilihan : 3

No      : 2
Nama    : popol
Nilai   : 78
Masukkan setelah nilai ? 1

Data dalam SLL :

No      = 1
Nama    = zai
Nilai   = 89

No      = 2
Nama    = popol
Nilai   = 78

No      = 4
Nama    = dan
Nilai   = 68

1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit
Masukkan pilihan : 4

No      : 3
Nama    : yeye
Nilai   : 88
Masukkan sebelum nilai ? 4

```

Data dalam SLL :

No = 1
Nama = zai
Nilai = 89

No = 2
Nama = popol
Nilai = 78

No = 3
Nama = yeye
Nilai = 88

No = 4
Nama = dan
Nilai = 68

1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit
Masukkan pilihan : 0

Tambahan :

untuk file run_multi.sh dan run_single.sh berfungsi untuk build and run percobaan percobaan diatas

file : run_multi.sh

```
#!/bin/bash

gcc -o tes *.c multi_function/*.c $(find ./single_function -name '*.c'
! -name 'main.c')
if [ $? -eq 0 ]; then
    ./tes
fi
```

disini kompiler akan mem-build semua file dalam folder 2.2 dan multi serta single kecuali file main.c dalam folder single_function.

file : run_single.sh

```
#!/bin/bash

gcc -o tes *.c single_function/*.c
if [ $? -eq 0 ]; then
    ./tes
fi
```

sedangkan untuk percobaan 1 sampai 4 kompiler akan mem-build hanya semua file pada folder 2.2 dan single function.

Apabila build berhasil maka akan lanjut dengan command ./tes untuk run program yang sudah di build.

(Linux environment only)