

27 Februari 2024

PRAKTIKUM 2.1

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

Sebelumnya saya klarifikasi terlebih dahulu, bahwa saya disini akan menggunakan modular programming. Jadi saya memisah file berdasarkan fungsi nya.

berikut link github untuk source code lengkapnya [disini](#)

Pertama saya membuat header file yang akan di include kelak dalam file lain lain. File ini berguna untuk mendeklarasikan preprocessor library, variabel, dan prototype fungsi. berikut adalah isi dari file “head.h”

a. head.h

```
#ifndef HEAD_H
#define HEAD_H

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

struct simpul
{
    int data;
    struct simpul *next;
};
typedef struct simpul Node;

extern Node *head;
extern Node *current;
extern int failedInBefore;
extern int failedInAfter;

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

#endif
```

Kedua dalam file “secondary.c” adalah file yang memuat fungsi sampingan untuk linked list, seperti alokasi, tampil, dan free_memory.

b. secondary.c

```
#include "head.h"

Node *head = NULL, *current;

void alokasi()
{
    current = (Node *)malloc(sizeof(Node));
    if (current == NULL)
    {
        printf("gagal mengalokasikan memori\n");
        exit(1);
    }
    printf("Nilai yang akan disimpan : ");
    scanf("%d", &current->data);
    getchar();
    current->next = NULL;
}

void tampil()
{
    Node *baca;
    baca = head;
    printf((failedInAfter == 1 || failedInBefore == 1) ? "gagal menemukan key, silahkan coba lagi\n" : "");
    printf("\nData dalam SLL :\n");
    while (baca != NULL)
    {
        printf("Nilai = %d\n", baca->data);
        baca = baca->next;
    }
    printf("\n");
}

void free_memory()
{
    Node *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
}

```

Ketiga dalam file insert awal, akhir, after, dan before saya membuat 2 section, pertama saat single (hanya menggunakan satu fungsi saja) dan yang kedua saat keempat fungsi tersebut digunakan secara terintegrasi dalam file “MenuInsert.c”. Maka ketika saat ingin menggunakan hanya ingin menggunakan satu fungsi saja maka kita tinggal *comment out* bagian multi function purposes dari part beginning sampai end, dan begitupun sebaliknya

1. INSERT AWAL (awal.c):

Source Code :

```

#include "head.h"
//comment out the unused part
void awal()
{
    printf("\nSLL - INSERT DI AWAL\n");
    //BEGINNING OF SINGLE FUNCTION PURPOSES
    int choice;
    do
    {
        alokasi();
        if (head != NULL)
        {
            current->next = head;
        }
        head = current;
        printf("Lagi (y/t)? ");
    }
}

```

```

        choice = getchar();
    } while (choice == 'y' || choice == 'Y');
    tampil();
    free_memory();
    exit(1);
//END OF SINGLE FUNCTION PURPOSES

//BEGINNING OF MULTI FUNCTION PURPOSES
alokasi();
if (head != NULL)
{
    current->next = head;
}
head = current;
printf("Lagi (y/t)? ");
choice = getchar();
//END OF MULTI FUNCTION PURPOSES
}

```

Output :

```

SLL - INSERT DI AWAL
Nilai yang akan disimpan : 3
Lagi (y/t)? y
Nilai yang akan disimpan : 5
Lagi (y/t)? y
Nilai yang akan disimpan : 19
Lagi (y/t)? y
Nilai yang akan disimpan : 14
Lagi (y/t)? t

Data dalam SLL :
Nilai = 14
Nilai = 19
Nilai = 5
Nilai = 3

```

2. INSERT AKHIR (akhir.c)

Source Code :

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

```

//comment out the unused part
void akhir()
{
    Node *tail;
    printf("\nSLL - INSERT DI AKHIR\n");
    // BEGINNING OF SINGLE FUNCTION PURPOSES
    int choice;
    do
    {
        alokasi();
        if (head == NULL)
            head = current;
        else
        {
            tail = head;
            while (tail->next != NULL)
                tail = tail->next;
            tail->next = current;
        }
        choice = getchar();
    } while (choice == 'y' || choice == 'Y');
    tampil();
    free_memory();
    exit(1); //comment out exit when using after or before
function! (single function purposes)

    // END OF SINGLE FUNCTION PURPOSES

    //BEGINNING OF MULTI FUNCTION PURPOSES
    alokasi();
    if (head == NULL)
        head = current;
    else
    {
        tail = head;
        while (tail->next != NULL)
            tail = tail->next;
        tail->next = current;
    }
    //END OF MULTI FUNCTION PURPOSES
}

```

Output :

```
SLL - INSERT DI AKHIR
Nilai yang akan disimpan : 4
Lagi (y/t)? y
Nilai yang akan disimpan : 7
Lagi (y/t)? y
Nilai yang akan disimpan : 9
Lagi (y/t)? y
Nilai yang akan disimpan : 12
Lagi (y/t)? t
```

```
Data dalam SLL :
Nilai = 4
Nilai = 7
Nilai = 9
Nilai = 12
```

3. INSERT AFTER (after.c)

Source Code :

```
#include "head.h"
// comment out the unused part
int failedInAfter = 0;
void after()
{
    Node *findKey;
    int key;
    printf("\nSLL - INSERT DI AFTER\n");
    // BEGINNING OF SINGLE FUNCTION PURPOSES

    akhir();
    alokasi();
    printf("Masukkan setelah nilai ? ");
    scanf("%d", &key);
    getchar();

    findKey = head;
    while (findKey != NULL && findKey->data != key)
        findKey = findKey->next;

    if (findKey == NULL)
        failedInAfter = 1;
    else
```

```

    {
        current->next = findKey->next;
        findKey->next = current;
    }
    tampil();
    free_memory();
    exit(1);
    // END OF SINGLE FUNCTION PURPOSES

    // BEGINNING OF MULTI FUNCTION PURPOSES
    // alokasi();
    // printf("Masukkan setelah nilai ? ");
    // scanf("%d", &key);

    // findKey = head;
    // while (findKey != NULL && findKey->data != key)
    //     findKey = findKey->next;

    // if (findKey == NULL)
    //     failedInAfter = 1;
    // else
    // {
    //     current->next = findKey->next;
    //     findKey->next = current;
    // }
    // END OF MULTI FUNCTION PURPOSES
}

```

Output : (belum sesuai harapan, segera saya debug)

4. INSERT BEFORE (before.c)

Source Code :

```

#include "head.h"
// comment out the unused part
int failedInBefore = 0;
void before()
{
    Node *findKey, *prev;

    prev = NULL;
    findKey = head;

```



```

int key;
printf("\nSLL - INSERT BEFORE\n");
// BEGINNING OF SINGLE FUNCTION PURPOSES
akhir();
alokasi();
printf("Masukkan sebelum nilai ? ");
scanf("%d", &key);
getchar();

while (findKey != NULL && findKey->data != key)
{
    prev = findKey;
    findKey = findKey->next;
}
if (findKey == NULL)
    failedInBefore = 1;
else
{
    if (prev == NULL)
    {
        current->next = head;
        head = current;
    }
    else
    {
        current->next = prev->next;
        prev->next = current;
    }
}
tampil();
free_memory();
exit(1);
// END OF SINGLE FUNCTION PURPOSES

// BEGINNING OF MULTI FUNCTION PURPOSES
// alokasi();
// printf("Masukkan sebelum nilai ? ");
// scanf("%d", &key);

// while (findKey != NULL && findKey->data != key)
// {
//     prev = findKey;

```

```

//      findKey = findKey->next;
// }
// if (findKey == NULL)
//      failedInBefore = 1;
// else
// {
//      if (prev == NULL)
//      {
//          current->next = head;
//          head = current;
//      }
//      else
//      {
//          current->next = prev->next;
//          prev->next = current;
//      }
// }
// END OF MULTI FUNCTION PURPOSES
}

```

Output : (output belum sesuai harapan, segera saya debug)

5. MENU INSERT (MenuInsert.c)

Source Code :

```

#include "head.h"

int main(int argc, char const *argv[])
{
    int index = 0;
    int exit = 0;
    int insert_choice;
    while (!exit)
    {
        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();
                clearScreen();

```

```

        break;
    case 2:
        akhir();
        clearScreen();
        break;
    case 3:
        // comment out line 24 for multi function purpose
        // after();
        // comment out line 26 for single function purpose
        (head != NULL && index != 0) ? after(), clearScreen()
: clearScreen(), printf("SLL Masih kosong, tidak bisa insert
after\n");
        break;
    case 4:
        // comment out line 29 for multi function purpose
        // before();
        // comment out line 31 for single function purpose
        (head != NULL && index != 0) ? after(), clearScreen()
: clearScreen(), printf("SLL Masih kosong, tidak bisa insert
after\n");
        break;
    case 0:
        exit = 1;
        break;
    default:
        clearScreen();
        printf("Pilihan anda tadi tidak valid, silahkan coba
lagi\n");
        break;
}
tampil();
index++;
}
free_memory();
return 0;
}

```

Output :

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

```
SLL - INSERT DI AWAL
Nilai yang akan disimpan : 23
```

```
Data dalam SLL :
Nilai = 23
```

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

```
SLL - INSERT DI AKHIR
Nilai yang akan disimpan : 12
```

```
Data dalam SLL :
Nilai = 23
Nilai = 12
```

1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit

Masukkan pilihan : 3

SLL - INSERT DI AFTER

Nilai yang akan disimpan : 44

Masukkan setelah nilai ? 23

Data dalam SLL :

Nilai = 23

Nilai = 44

Nilai = 12

1. AWAL
2. AKHIR
3. AFTER
4. BEFORE
0. exit

Masukkan pilihan : 4

SLL - INSERT BEFORE

Nilai yang akan disimpan : 50

Masukkan sebelum nilai ? 44

Data dalam SLL :

Nilai = 23

Nilai = 50

Nilai = 44

Nilai = 12

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

```
Data dalam SLL :
Nilai = 23
Nilai = 50
Nilai = 44
Nilai = 12
```