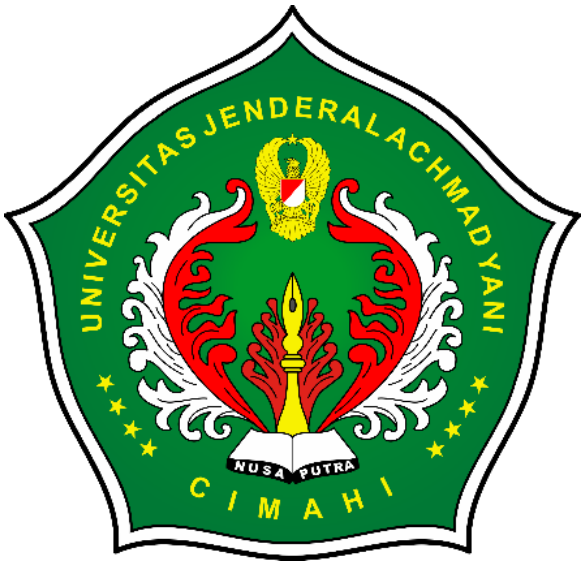
LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL 8
ADT Double Linked List

DISUSUN OLEH :

AJI KARTIKO HARTANTO - 2350081062



PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN INFORMATIKA

UNIVERSITAS JENDERAL ACHMAD YANI

TAHUN 2024

DAFTAR ISI

DAFTAR GAMBAR

BAB I. HASIL PRAKTIKUM

I.1 Latihan

I.1.A. Source Code Boolean.h

```
/*
    Program     : boolean.h
    Deskripsi   : Header file dari boolean
*/

#ifndef boolean_H
#define boolean_H
#define true 1
#define false 0
#define boolean unsigned char

#endif
```

I.1.B. Source Code doubleLinkedList.h

```
/*
    Program     : doubleLinkedList.h
    Author      : 2350081062, Aji Kartiko Hartanto
    Kelas       : C
    Deskripsi   : Header file dari prototype double linked list
    Tanggal     : 05-06-2024
*/

#ifndef DoubleLinkedList_H
#define DoubleLinkedList_H

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

#define Nil NULL
#define Info(P) (P)->info
```

```
#define Next(P) (P)->next
#define Prev(P) (P)->prev
#define First(L) (L).First

typedef int infoType;
typedef struct tElmList *address;

typedef struct tElmList {
    infoType info;
    address next;
    address prev;
} ElmList;

typedef struct {
    address First;
} List;

boolean ListEmpty(List L);

void CreateList(List *L);

address Alokasi(infoType X);

void Dealokasi(address P);

address Search(List L, infoType X);

void AddFirst(List *L, infoType X);

void AddLast(List *L, infoType X);

void DelFirst(List *L, infoType *X);

void DelLast(List *L, infoType *X);
void CetakList(List L);
#endif
```

I.1.C. Source Code doubleLinkedList.c

```
/*
    Program     : doubleLinkedList.c
    Author      : 2350081062, Aji Kartiko Hartanto
    Kelas       : C
    Deskripsi   : body file dari prototype double linked list
    Tanggal     : 05-06-2024
 */

#include "doubleLinkedList.h"

boolean ListEmpty(List L) {
    if(First(L) == Nil) {
        return true;
    } else {
        return false;
    }
}

void CreateList(List *L) {
    First(*L) = Nil;
}

address Alokasi(infoType X) {
    // kamus
    address newNode;

    // alokasi
    newNode = (ElmList *) malloc(sizeof(ElmList));

    // algoritma
    Info(newNode) = X;
    Next(newNode) = Nil;
    Prev(newNode) = Nil;

    return newNode;
}
```

```
void Dealokasi(address P) {
    free(P);
}

address Search(List L, infoType X) {
    address current;

    current = First(L);
    while (current != Nil) {
        if (Info(current) == X) {
            return current;
        }

        current = Next(current);
    }

    return Nil;
}

void AddFirst(List *L, infoType X) {
    address newNode;

    newNode = Alokasi(X);

    if(newNode != Nil) {
        Info(newNode) = X;
        Next(newNode) = First(*L);
        First(*L) = Prev(newNode);
        First(*L) = newNode;
    }
}

void AddLast(List *L, infoType X) {
    address newNode, current;

    newNode = Alokasi(X);
```

```
    current = First(*L);

    if (newNode != Nil) {
        while (Next(current) != Nil) {
            current = Next(current);
        }

        Next(current) = newNode;
        Prev(newNode) = current;
    }
}

void DelFirst(List *L, infoType *X) {
    address delNode, tmp;

    delNode = First(*L);
    First(*L) = Next(First(*L));

    *X = Info(delNode);
    tmp = delNode;
    Dealokasi(tmp);

    tmp = Nil;
    Prev(First(*L)) = Nil;
}

void DelLast(List *L, infoType *X) {
    address delNode, tmp, tmp2;

    delNode = First(*L);
    *X = Info(delNode);
    tmp = (delNode);

    while(Next(tmp) != Nil) {
        tmp = Next(tmp);
    }
```

```
    tmp2 = Prev(tmp);
    Next(tmp2) = Nil;
    Dealokasi(tmp);
}

void CetakList(List L) {
    address current;

    current = First(L);
    printf("Null");
    while (current != Nil) {
        printf(" <- [%d] -> ", Info(current));

        current = Next(current);
    }

    printf("Null");
}
```

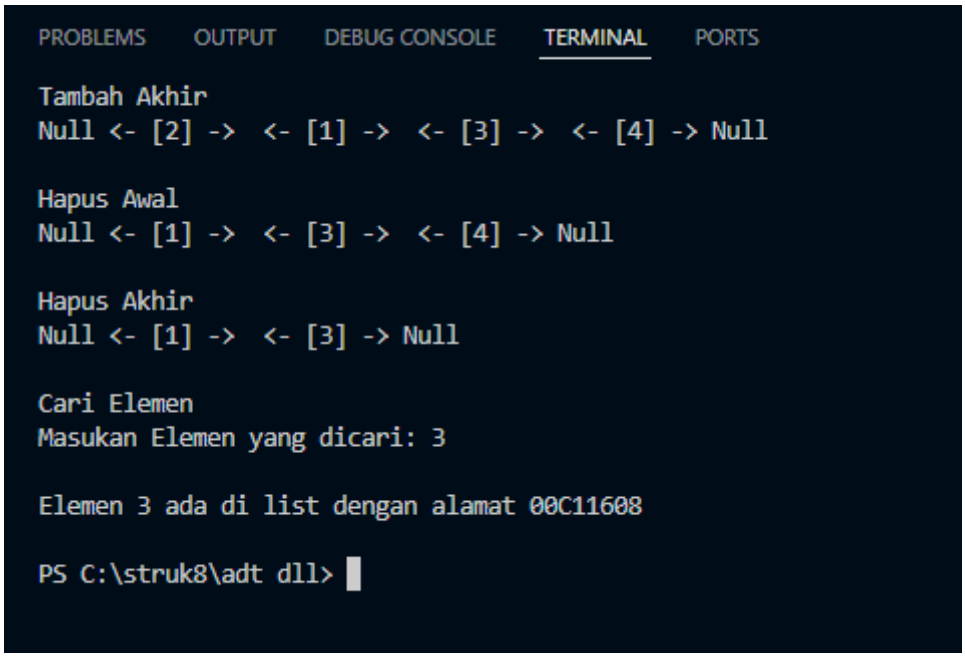
I.1.D. Source Code mdoubleLinkedList.c

```
/*
    Program     : mdoubleLinkedList.c
    Author      : 2350081062, Aji Kartiko Hartanto
    Kelas       : C
    Deskripsi   : main program dari Adt double linked list
    Tanggal     : 05-06-2024
*/

#include "doubleLinkedList.c"

int main() {
    List node;
    infoType info, elemen;
    address Tmp;

    CreateList(&node);
```

```
    printf("\nApakah List masih kosong?\n");
    if(ListEmpty(node)) {
        printf("List masih kosong\n");
    } else {
        printf("List tidak kosong\n");
    }

    printf("\n\nTambah Awal\n");
    AddFirst(&node, 1);
    AddFirst(&node, 2);
    CetakList(node);

    printf("\n\nTambah Akhir\n");
    AddLast(&node, 3);
    AddLast(&node, 4);
    CetakList(node);

    printf("\n\nHapus Awal\n");
    DelFirst(&node, &elemen);
    CetakList(node);

    printf("\n\nHapus Akhir\n");
    DelLast(&node, &elemen);
    CetakList(node);

    printf("\n\nCari Elemen\n");
    printf("Masukan Elemen yang dicari: ");
    scanf("%d", &info);

    Tmp = Search(node, info);
    if (Tmp != Nil) {
        printf("\nElemen %d ada di list dengan alamat %p", info,
Tmp);
    } else {
        printf("Elemen %d tidak ada di list", info);
    }
```

```
    printf("\n\n");
    return 0;
}
```

I.1.E. Hasil



Gambar I.1 Output Program mdoubleLinkedList.c

I.1.F. Analisa

Dengan menggunakan bahasa C, program ini menjalankan operasi-operasi dasar pada list ganda terurut. Fungsi ListEmpty memeriksa apakah list kosong dengan melihat apakah elemen pertamanya adalah Nil, mengembalikan benar jika list kosong, dan salah jika tidak. Fungsi CreateList menginisialisasi list dengan mengatur elemen pertamanya menjadi Nil. Fungsi Alokasi mengatur memori untuk node baru, mengisi informasi node, dan mengatur pointer Next dan Prev menjadi Nil.

Fungsi AddFirst menambahkan node baru di awal list, sedangkan Fungsi DelFirst menambahkan node baru di akhir list. Fungsi DelFirst menghapus node pertama dari list dan mengembalikan nilai node tersebut, serta mengatur pointer elemen pertama baru dan Prev elemen pertama menjadi Nil. Fungsi DelLast menghapus node terakhir dari list dengan mengikuti pointer Next hingga mencapai node terakhir, mengatur pointer Next node sebelumnya menjadi Nil, dan membebaskan memori node terakhir.

BAB II. KESIMPULAN

Secara keseluruhan, program ini menawarkan pemahaman dasar tentang struktur data list ganda terurut dan pengelolaan memori dinamis dalam C. Fungsi-fungsinya menunjukkan cara menginisialisasi list, menambah dan menghapus elemen, dan menjaga integritas list, termasuk pengaturan pointer yang tepat.