

# LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Struktur Data  
Kelas : 2IA22  
Praktikum Ke- : 2  
Tanggal : 4 November 2023  
Materi : Double Linked List  
NPM : 51422041  
Nama : Muhammad Faris Rasyid R  
Ketua Asisten : Muhammad Imam Prasetyo  
Nama Asisten :  
Paraf Asisten :  
Jumlah Lembar : 5 Lembar



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

## LISTING

```
index.cpp

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

struct node {
    struct node *prev;
    struct node *next;
    int data;
};

struct node *head = NULL;

void tambahDataAwal();
void tambahDataAkhir();
void tampil();

int main() {
    int pilih = 0;
    while (pilih != 9) {
        printf("\n***** Main Menu *****\n");
        printf("\nPilihlah opsi manasaja dari pilihan dibawah ini: \n");
        printf("\n*****\n");
        printf("\n1. Masukkan data di awal\n2. Masukkan data di akhir\n3. Tampilkan data");
        printf("\nMasukan Pilihan Anda: ");
        scanf("%d", &pilih);
        switch (pilih) {
            case 1:
                tambahDataAwal();
                break;
            case 2:
                tambahDataAkhir();
                break;
            case 3:
                tampil();
                break;
            default:
                printf("Tolong masukkan pilihan yang sesuai ");
        }
    }
    return 0;
}
```



C++ index.cpp

```
void tambahDataAwal() {
    struct node *ptr;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("\nAlokasi memori gagal");
    } else {
        printf("\nMasukkan data berupa angka: ");
        scanf("%d", &item);
        ptr->data = item;
        if (head == NULL) {
            ptr->next = NULL;
            ptr->prev = NULL;
            head = ptr;
        } else {
            ptr->next = head;
            ptr->prev = NULL;
            head->prev = ptr;
            head = ptr;
        }
        printf("\nData Berhasil dimasukkan\n");
    }
}
```



CodeImage

```
void tambahDataAkhir() {
    struct node *ptr, *temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("\nAlokasi memori gagal");
    } else {
        printf("\nMasukkan data berupa angka: ");
        scanf("%d", &item);
        ptr->data = item;
        if (head == NULL) {
            ptr->next = NULL;
            ptr->prev = NULL;
            head = ptr;
        } else {
            temp = head;
            while (temp->next != NULL) {
                temp = temp->next;
            }
            temp->next = ptr;
            ptr->prev = temp;
            ptr->next = NULL;
        }
        printf("\nData berhasil dimasukkan\n");
    }
}

void tampil() {
    struct node *ptr;
    printf("\nList: \n");
    ptr = head;
    while (ptr != NULL) {
        printf("%d\n", ptr->data);
        ptr = ptr->next;
    }
}
```

## LOGIKA

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

struct node {
    struct node *prev;
    struct node *next;
    int data;
};

struct node *head = NULL;

void tambahDataAwal();
void tambahDataAkhir();
void tampil();
```

Bagian ini adalah pengaturan awal program C yang menggunakan dua library standar, yaitu ``<stdio.h>`` dan ``<stdlib.h>``:

1. `##include <stdio.h>`: Ini adalah preprocessor directive (direktif prapemrosesan) yang mengimpor library standar C yang disebut ``<stdio.h>``. Library ini digunakan untuk fungsi-fungsi dasar input dan output, seperti ``printf`` dan ``scanf``.
2. `##include <stdlib.h>`: Ini adalah preprocessor directive yang mengimpor library standar C ``<stdlib.h>``. Library ini digunakan untuk fungsi-fungsi yang berhubungan dengan alokasi memori dinamis dan pengelolaan memori, seperti ``malloc`` yang digunakan untuk mengalokasikan memori untuk node dalam linked list.

Selanjutnya, kode mendefinisikan sebuah struktur bernama ``node``:

- Struktur ``node`` digunakan untuk merepresentasikan setiap simpul dalam linked list. Setiap simpul memiliki tiga anggota:
  - ``struct node *prev``: Pointer yang menunjuk ke simpul sebelumnya dalam linked list. Dalam linked list g (doubly linked list), ini memungkinkan untuk navigasi maju dan mundur.
  - ``struct node *next``: Pointer yang menunjuk ke simpul berikutnya dalam linked list.
  - ``int data``: Bagian ini menyimpan data dalam simpul, dalam hal ini, tipe data integer.

Kemudian, program mendeklarasikan variabel ``head`` yang digunakan untuk menyimpan alamat dari simpul pertama dalam linked list. Awalnya, ``head`` diatur menjadi ``NULL`` untuk menunjukkan bahwa linked list masih kosong.

Selanjutnya, program mendeklarasikan tiga fungsi:

1. `void tambahDataAwal()`: Fungsi ini digunakan untuk menambahkan data di awal linked list.
2. `void tambahDataAkhir()`: Fungsi ini digunakan untuk menambahkan data di akhir linked list.
3. `void tampil()`: Fungsi ini digunakan untuk menampilkan seluruh data dalam linked list.

Fungsi-fungsi ini akan digunakan untuk mengelola linked list dan melakukan operasi seperti penambahan data dan penampilan data.

```
int main() {
    int pilih = 0;
    while (pilih != 9) {
        printf("\n***** Main Menu *****\n");
        printf("\nPilihlah opsi manasaja dari pilihan dibawah ini: \n");
        printf("\n*****\n");
        printf("\n1. Masukkan data di awal\n2. Masukkan data di akhir\n3. Tampilkan data");
        printf("\nMasukan Pilihan Anda: ");
        scanf("%d", &pilih);
        switch (pilih) {
            case 1:
                tambahDataAwal();
                break;
            case 2:
                tambahDataAkhir();
                break;
            case 3:
                tampil();
                break;
            default:
                printf("Tolong masukkan pilihan yang sesuai ");
        }
    }
    return 0;
}
```

Fungsi main() adalah titik awal eksekusi program. Dalam loop while, program menampilkan menu pilihan kepada pengguna. Pilihan menu termasuk menambahkan data di awal, menambahkan data di akhir, dan menampilkan data. Program akan terus berjalan hingga pengguna memilih untuk keluar dengan memasukkan angka 9.

```

void tambahDataAwal() {
    struct node *ptr;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("\nAlokasi memori gagal");
    } else {
        printf("\nMasukkan data berupa angka: ");
        scanf("%d", &item);
        ptr->data = item;
        if (head == NULL) {
            ptr->next = NULL;
            ptr->prev = NULL;
            head = ptr;
        } else {
            ptr->next = head;
            ptr->prev = NULL;
            head->prev = ptr;
            head = ptr;
        }
        printf("\nData Berhasil dimasukkan\n");
    }
}

```

Fungsi ini digunakan untuk menambahkan data di awal linked list. Pertama, ia mengalokasikan memori untuk node baru. Kemudian, meminta pengguna memasukkan data. Jika linked list masih kosong, node baru ini menjadi node pertama (head). Jika linked list tidak kosong, node baru ditambahkan di awal dan dihubungkan ke head sebelumnya.

```

void tambahDataAkhir() {
    struct node *ptr, *temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("\nAlokasi memori gagal");
    } else {
        printf("\nMasukkan data berupa angka: ");
        scanf("%d", &item);
        ptr->data = item;
        if (head == NULL) {
            ptr->next = NULL;
            ptr->prev = NULL;
            head = ptr;
        } else {
            temp = head;
            while (temp->next != NULL) {
                temp = temp->next;
            }
            temp->next = ptr;
            ptr->prev = temp;
            ptr->next = NULL;
        }
        printf("\nData berhasil dimasukkan\n");
    }
}

```

: Fungsi ini digunakan untuk menambahkan data di akhir linked list. Ia mengalokasikan memori untuk node baru, meminta pengguna memasukkan data, dan menambahkan node baru setelah node terakhir dalam linked list.

```

void tampil() {
    struct node *ptr;
    printf("\nList: \n");
    ptr = head;
    while (ptr != NULL) {
        printf("%d\n", ptr->data);
        ptr = ptr->next;
    }
}

```

Fungsi ini digunakan untuk menampilkan semua data dalam linked list. Ia menggunakan pointer ptr untuk iterasi melalui linked list, mulai dari head, dan mencetak data pada setiap simpul.



## OUTPUT

```
index.sh

cd "/mnt/AE56E58256E54C21/about faris/FTI/Tk2/Praktikum/labti/STRUKTUR_DATA/pert3/output"
./"pertiga"
[oh-my-zsh] It's time to update! You can do that by running `omz update`

FTI on  main [!?]
> cd "/mnt/AE56E58256E54C21/about faris/FTI/Tk2/Praktikum/labti/STRUKTUR_DATA/pert3/output"

STRUKTUR_DATA/pert3/output on  main [!?]
> ./"pertiga"

***** Main Menu *****

Pilihlah opsi manasaja dari pilihan dibawah ini:

*****

1. Masukkan data di awal
2. Masukkan data di akhir
3. Tampilkan data
Masukan Pilihan Anda: 1

Masukkan data berupa angka: 3

Data Berhasil dimasukkan

***** Main Menu *****

Pilihlah opsi manasaja dari pilihan dibawah ini:

*****

1. Masukkan data di awal
2. Masukkan data di akhir
3. Tampilkan data
Masukan Pilihan Anda: 2

Masukkan data berupa angka: 4

Data berhasil dimasukkan

***** Main Menu *****

Pilihlah opsi manasaja dari pilihan dibawah ini:

*****

1. Masukkan data di awal
2. Masukkan data di akhir
3. Tampilkan data
Masukan Pilihan Anda: 3

List:
3
4
```

