

Modul Praktikum 3 : Double Linked List

Kompetensi:

Mahasiswa mampu mengimplementasikan struktur data doublelinked list.

Tujuan Instruksional Khusus:

Mahasiswa mampu mengimplementasikan struktur data single linked list, dengan operasi meliputi:

1. Pembentukan node awal
2. Penambahan node pada posisi depan, tengah, dan belakang.
3. Penghapusan node pada posisi depan, tengah, dan belakang.
4. Pengaksesan data dari depan (head ke tail)
5. Pengaksesan data dari belakang (tail ke head)

Praktikum:

1. Peralatan
 - a. Perangkat komputer
 - b. Sistem Operasi Windows
 - c. C++ Compiler (MinGW)
 - d. Text Editor (Geany, Notepad++, Visual Studio Code, SublimeText, dll)
2. Prosedur
 - a. Baca dan pahami semua tahapan praktikum dengan cermat.
 - b. Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
 - c. Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
 - d. Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
 - e. Pastikan diri anda tidak menyentuh sumber listrik.
3. Teori Double Linked List

Struktur Node :

```
struct Node{
    int data;
    Node *next;
    Node *prev;
};

Node *n=NULL, *head=NULL, *tail=NULL, *x=NULL;
```

Pembuatan Node Awal:

```
void buatNodeBaru(int i){
    n = new Node;
    n->data = i;
    n->prev = NULL;
    head = n;
    tail = n;
    tail->next = NULL;
}
```

Penambahan node pada posisi belakang:

```
void tambahDiBelakang(int i){
    n = new Node;
    n->data = i;
    n->prev = tail;
    tail->next = n;
    tail = n;
    tail->next = NULL;
}
```

Penambahan node pada posisi depan:

```
void tambahDiDepan(int i){
    n = new Node;
    n->data = i;
    n->next = head;
    head->prev = n;
    n->prev = NULL;
    head = n;
}
```

Penambahan node pada posisi tengah:

```
void tambahDiTengah(int i, int j){
    x = head;
    while(x->data != j) x=x->next;
    n = new Node;
    n->data = i;
    n->next = x->next;
    x->next = n;
    n->prev = x;
    x = n->next;
    x->prev = n;
}
```

Penghapusan node pada posisi depan:

```
void hapusDiDepan(){
    x = head;
    head = x->next;
    head->prev = NULL;
    delete(x);
    x = NULL;
}
```

Penghapusan node pada posisi belakang:

```
void hapusDiBelakang(){
    x = tail;
    tail = tail->prev;
    tail->next = NULL;
    delete(x);
    x = NULL;
}
```

Penghapusan node pada posisi tengah:

```
void hapusDiTengah(int i){
    n = NULL;
    x = head;
    while(x->data != i){
        n = x;
        x = x->next;
    }
    n->next = x->next;
    x = x->next;
    delete(x->prev);
    x->prev = n;
}
```

Pengaksesan data dari depan:

```
void tampilDariDepan(){
    x = head;
    while(x != NULL){
        cout << x->data << " ";
        x = x->next;
    }
}
```

Pengaksesan data dari belakang:

```
void tampilDariBelakang(){
    x = tail;
    while(x != NULL){
        cout << x->data << " ";
        x = x->prev;
    }
}
```

4. Kegiatan Praktikum:

Buat program untuk implementasi single linked list. Output dapat dilihat pada Gambar di bawah.



Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.