

Modul Praktikum 2 : Single Linked List

Kompetensi:

Mahasiswa mampu mengimplementasikan Stack dan Queue menggunakan Struktur Data Sederhana.

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.

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

a. Stack

Struktur Node :

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

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

Pembuatan Node Awal:

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

Penambahan node pada posisi belakang:

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

Penghapusan node pada posisi depan:

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

Penghapusan node pada posisi belakang:

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

Penghapusan node pada posisi tengah:

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

Tampil data:

```
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;
}
```

Penambahan node pada posisi tengah:

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

4. Kegiatan Praktikum:

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



```
C:\Windows\SYSTEM32\cmd.exe
Menu:
1. Buat Node Awal
2. Tambah Node
3. Hapus Node
4. Tampil Data
5. Keluar
Masukkan pilihan [1..5] :
```

```
C:\Windows\SYSTEM32\cmd.exe
Buat Node Baru
=====
Masukkan nilai: 10
```

```
C:\Windows\SYSTEM32\cmd.exe
Buat Node Baru
=====
Masukkan nilai: 10
Nilai 10 berhasil dimasukkan di node awal.
```

```
C:\Windows\SYSTEM32\cmd.exe
Isi Linked List
=====
10
```

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