

Laporan Praktikum 8

Struktur Data Algoritma



Materi "Single Linked List"

**Nama :
Muhammad Azhar Rasyad**

**NIM :
0110217029**

**Program Studi :
Teknik Informatika 1**

Single Linked List

Single Linked List merupakan proses dari node head kemudian disambung dengan node lain hingga node tail secara searah sehingga pointer yang digunakan hanya 1 dan terus berlanjut dari node head ke node tail.

Berikut contoh **ilustrasi** single linked list :



Berikut contoh **source code** single linked list :

```
#include <iostream>

using namespace std;

struct node
{
    int data;
    node *next;
} *tail = NULL, *head = NULL, *baru, *bantu;

void membuatNode(int nilai)
{
    baru = new node;
    baru -> data = nilai;
    baru -> next = NULL;
}

int nilai;

void tambah()
{
    cout << "\nMasukkan Nilai = ";
    cin >> nilai;
    membuatNode(nilai);

    if(head == NULL)
    {
        head = baru;
        tail = baru;
        tail -> next = NULL;
    }
    else
    {

```

```

        baru -> next = head;
        head = baru;
    }
    cout << "Data Berhasil Ditambahkan";
}

```

```

void lihat()
{
    if(head == NULL)
    {
        cout << "\nKosong Bro...";
    }
    else if(head -> next = NULL)
    {
        cout << head -> data;
    }
    else
    {
        cout << "\nLinked List Berisi : ";
        bantu = head;
        while(bantu != NULL)
        {
            cout << bantu -> data << " ";
            bantu = bantu -> next;
        }
    }
}

```

```

void hapus()
{
    if(head != NULL)
    {
        if(head -> next = NULL)
        {
            head = NULL;
            tail = NULL;
            cout << "\nData Berhasil Dihapus";
        }
        else
        {
            bantu = head;
            bantu = bantu -> next;
            head -> next = NULL;
            head = bantu;
            cout << "\nData Berhasil Dihapus";
        }
    }
    else
    {
        cout << "\nKosong Bro...";
    }
}

```

```
    }  
}
```

```
int main()  
{  
    int pilih;
```

```
    do  
    {  
        cout << "1. Tambah \n";  
        cout << "2. Lihat \n";  
        cout << "3. Hapus \n";  
        cout << "4. Keluar \n";  
        cout << "Silahkan Pilih = ";  
        cin >> pilih;
```

```
        switch(pilih)  
        {  
            case 1 :  
                tambah();  
                cout << endl << endl;  
                break;
```

```
            case 2 :  
                lihat();  
                cout << endl << endl;  
                break;
```

```
            case 3 :  
                hapus();  
                cout << endl << endl;  
                break;
```

```
            case 4 :  
                cout << "\nBye..." << endl;  
                break;
```

```
        }  
    }  
    while(pilih < 4);  
}
```

Berikut **penjelasan** dari source code diatas :

```
1  #include <iostream>
2
3  using namespace std;
4
```

Code diatas merupakan **standar** dari **bahasa pemrograman C++** yang berfungsi untuk **memunculkan library C++** agar dapat input output dan lainnya dengan perintah tersebut.

```
5  struct node
6  {
7      int data;
8      node *next;
9  } *tail = NULL, *head = NULL, *baru, *bantu;
10
```

Code diatas digunakan untuk **mendeklarasikan sebuah node** pada linked list dengan **menggunakan struct** agar isi dari node tersebut **memiliki tipe data yang bermacam-macam**. Dalam struct tersebut minimal harus memiliki 2 field yaitu **pertama untuk data** dan **kedua untuk alamatnya** agar setiap node memiliki alamatnya masing-masing serta **setiap linked list minimal memiliki alamat head dan tail**.

```
11 void membuatNode(int nilai)
12 {
13     baru = new node;
14     baru -> data = nilai;
15     baru -> next = NULL;
16 }
17
```

Code diatas berfungsi **membuat sebuah node** dalam suatu linked list serta mengisi setiap fieldnya dengan data yang diperlukan.

```

18  int nilai;
19
20  void tambah()
21  {
22      cout << "\nMasukkan Nilai = ";
23      cin >> nilai;
24      membuatNode(nilai);
25
26      if(head == NULL)
27      {
28          head = baru;
29          tail = baru;
30          tail -> next = NULL;
31      }
32      else
33      {
34          baru -> next = head;
35          head = baru;
36      }
37      cout << "Data Berhasil Ditambahkan";
38  }
39

```

Code diatas berfungsi **menambahkan data** yang dimasukkan ke dalam node dengan 2 kondisi yaitu :

- **Kondisi pertama** jika head berisi NULL **artinya tidak memiliki node** maka head dan tail dibuat serta beralamat pada alamat yang sama yaitu node pertama.
- **Kondisi kedua** jika head memiliki node maka node yang telah ditambahkan mempunyai alamat head menggantikan alamat node yang sudah terisi karena node baru diletakkan di awal.

```

40 void lihat()
41 {
42     if(head == NULL)
43     {
44         cout << "\nKosong Bro...";
45     }
46     else if(head -> next == NULL)
47     {
48         cout << head -> data;
49     }
50     else
51     {
52         cout << "\nLinked List Berisi : ";
53         bantu = head;
54         while(bantu != NULL)
55         {
56             cout << bantu -> data << " ";
57             bantu = bantu -> next;
58         }
59     }
60 }
61

```

Code diatas berfungsi untuk **melihat data** pada node dengan ada 3 kondisi yaitu :

- **Kondisi pertama** jika tidak ada node maka akan memunculkan notifikasi bahwa tidak ada node yang tersedia untuk dilihat datanya.
- **Kondisi kedua** jika hanya ada 1 node yang tersedia maka hanya akan menampilkan data pada 1 node.
- **Kondisi ketiga** jika node yang tersedia lebih dari maka akan menampilkan semua node yang ada.

```

62 void hapus()
63 {
64     if(head != NULL)
65     {
66         if(head -> next = NULL)
67         {
68             head = NULL;
69             tail = NULL;
70             cout << "\nData Berhasil Dihapus";
71         }
72     else
73     {
74         bantu = head;
75         bantu = bantu -> next;
76         head -> next = NULL;
77         head = bantu;
78         cout << "\nData Berhasil Dihapus";
79     }
80 }
81 else
82 {
83     cout << "\nKosong Bro...";
84 }
85 }
86

```

Code diatas digunakan untuk **menghapus node** dengan ada 2 kondisi yaitu :

- **Kondisi pertama** jika terdapat node 1 atau lebih dari 1 maka node akan dihapus.
- **Kondisi kedua** jika tidak ada node maka akan menampilkan notifikasi bahwa tidak ada node yang tersedia untuk dihapus.


```

87  int main()
88  {
89      int pilih;
90      do
91      {
92          cout << "1. Tambah \n";
93          cout << "2. Lihat \n";
94          cout << "3. Hapus \n";
95          cout << "4. Keluar \n";
96          cout << "Silahkan Pilih = ";
97          cin >> pilih;
98          switch(pilih)
99          {
100             case 1 :
101                 tambah();
102                 cout << endl << endl;
103                 break;
104             case 2 :
105                 lihat();
106                 cout << endl << endl;
107                 break;
108             case 3 :
109                 hapus();
110                 cout << endl << endl;
111                 break;
112             case 4 :
113                 cout << "\nBye..." << endl;
114                 break;
115             }
116         }
117         while(pilih < 4);
118     }

```

Code diatas berfungsi sebagai **program utama** supaya prosedur-prosedur diatas dapat diimplementasikan kedalam sebuah program C++ serta menggunakan proses switch case dengan mengulang agar linked list dapat dimodifikasi sesuai kebutuhan.

Berikut **hasil kompilasi** dari source code diatas :

```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = █
```

Gambar diatas merupakan tampilan awal saat memasuki program dan terdapat 4 menu yang bisa dilihat diatas kemudian dipilih sesuai keperluan, berikut penjelasannya :

- **Tambah** untuk menambahkan node kedalam linked list
- **Lihat** untuk melihat node didalam linked list
- **Hapus** untuk menghapus node didalam linked list
- **Keluar** untuk keluar dari program linked list

Jika tidak memilih pilihan dari 1 sampai 4 maka otomatis akan keluar dari program.

```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 1
Masukkan Nilai = █
```

Gambar diatas merupakan contoh **fungsi dari menu ke 1** yaitu **menambahkan node**, jika memilih menu ke 1 maka **program akan meminta nilai** yang ingin dimasukkan kedalam node.

```
mazharrasyad@Mazharrasyad: ~/Desktop
mazharrasyad@Mazharrasyad:~/Desktop$ ./start
1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 1
Masukkan Nilai = 10
Data Berhasil Ditambahkan
1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = █
```

Pada contoh kali ini data yang digunakan menggunakan tipe data integer maka **jika data yang dimasukkan berupa integer** maka **proses**

berhasil namun jika data yang dimasukkan berupa string maka program akan terjadi error sehingga harus mengulang program dari awal.

```
mazharrasyad@Mazharrasyad: ~/Desktop
Data Berhasil Ditambahkan

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 2

Linked List Berisi : 10

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = █
```

Contoh berikutnya yaitu mencoba fungsi menu ke 2 yaitu **melihat node**, pada gambar diatas terlihat node berisi nilai 10 karena sebelumnya telah ditambahkan dengan nilai 10 pada fungsi tambah.

```
mazharrasyad@Mazharrasyad: ~/Desktop
Linked List Berisi : 10

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 3

Data Berhasil Dihapus

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = █
```

Selanjutnya contoh dari fungsi menu ke 3 yaitu **menghapus node**, pada gambar diatas terlihat bahwa node berhasil dihapus dengan adanya notifikasi "Data Berhasil Dihapus".

```
mazharrasyad@Mazharrasyad: ~/Desktop
Data Berhasil Dihapus

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 2

Kosong Bro...

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = █
```

Karena sebelumnya node telah dihapus maka jika digunakan lagi fungsi menu ke 2 yaitu melihat node maka hasilnya adalah kosong atau tidak ada node yang ada pada linked list.

```
mazharrasyad@Mazharrasyad: ~/Desktop
Data Berhasil Dihapus

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 2

Kosong Bro...

1. Tambah
2. Lihat
3. Hapus
4. Keluar
Silahkan Pilih = 4

Bye...
mazharrasyad@Mazharrasyad:~/Desktop$ █
```

Fungsi menu ke 4 yaitu keluar dari program linked list.

Kesimpulan

Single linked list merupakan kumpulan data yang disebut sebagai node, dari node tersebut kemudian disambungkan dengan node lainnya namun harus disambungkan secara berurutan dengan menggunakan alamat pointer next.