Laprak Struktur Data

Pertemuan 5

SINGLE LINKED LIST (BAGIAN KEDUA)

Disusun Oleh:

Dwi Candra Pratama/2211104035

AsPrak:

Dosen Pengampu:

Wahyu Andi Saputra S. Kom. MSc

PROGRAM STUDI REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

GUIDED

TUJUAN PRAKTIKUM

1. Memahami penggunaan linked list dengan pointer operator- operator dalam program.
2. Memahami operasi-operasi dasar dalam linked list.
3. Membuat program dengan menggunakan linked list dengan prototype yang ada

1. Searching

Searching merupakan operasi dasar list dengan melakukan aktivitas pencarian terhadap node tertentu. Proses ini berjalan dengan mengunjungi setiap node dan berhenti setelah node yang dicari ketemu. Dengan melakukan operasi searching, operasi-operasi seperti insert after, delete after, dan update akan lebih mudah. Semua fungsi dasar diatas merupakan bagian dari ADT dari singgle linked list, dan aplikasi pada bahasa pemrograman Cp semua ADT tersebut tersimpan dalam file *.c dan file *.h. Berikut hasil ketik melakukan Praktikum:

SourceCodenya:

```
#include <iostream>
using namespace std;

// Struktur unutk node dalam linked list
struct Node {
    int data;
    Node* next;
};

// Fungsi untuk menambahkan element baru ke awal linked list
void insertFirst(Node*& head, Node*& tail, int new_data) {
    Node* new_node = new Node();
    new_node->data = new_data;
    new_node->next = head;
    head = new_node;

    if (tail == nullptr) {
        tail = head;
    }
}

//Fungsi untuk menambahkan element baru ke akhir linked list
void insertLast(Node*& head, Node*& tail, int new_data) {
    Node* new_node = new Node();
    new_node->data = new_data;
    new_node->next = nullptr;

    if (head == nullptr) {
        head = new_node;
```

```
        tail = new_node;
    } else {
        tail->next = new_node;
        tail = new_node;
    }
}

// Fungsi untuk mencari elemen dalam linked list
int findElement(Node* head, int x) {
    Node* current = head;
    int index = 0;

    while (current != nullptr) {
        if (current->data == x) {
            return index;
        }
        current = current->next;
        index++;
    }
    return -1;
}

// Fungsi untuk menampilkan element dalam linked list
void display(Node* node) {
    while (node != nullptr) {
        cout << node->data << " ";
        node = node->next;
    }
    cout << endl;
}

// Fungsi untuk menghapus element dari linked list
void deleteElement(Node*& head, int x) {
    if (head == nullptr) {
        cout << "Linked list kososng" << endl;
        return;
    }

    if (head->data == x) {
        Node* temp = head;
        head = head->next;
        delete temp;
        return;
    }

    Node* current = head;
    while (current->next !=nullptr) {
        if (current->next->data == x) {
            Node* temp = current->next;
            current->next = current->next->next;
```

```
            delete temp;
            return;
        }
        current = current->next;
    }
}

int main() {
    Node* head = nullptr;
    Node* tail = nullptr;

    insertFirst(head, tail, 3);
    insertFirst(head, tail, 5);
    insertFirst(head, tail, 7);

    insertLast(head, tail, 11);
    insertLast(head, tail, 14);
    insertLast(head, tail, 18);

    cout << "Elemen dalam Linked List: ";
    display(head);

    int x;
    cout << "Masukkan elemen yang ingin dicari: ";
    cin >> x;

    int result = findElement(head, x);

    if (result == -1)
        cout << " Elemen tidak ditemukan dalam linked list" << endl;
    else
        cout <<" Elemen ditemukan pada indeks " << result << endl;

    cout << "Masukkan elemen yang ingin dicari: ";
    cin >> x;
    deleteElement(head, x);

    cout << "Elemen dalam linked list setelah penghapusan: ";
    display(head);

    return 0;
}
```

```
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan 5> cd "d:\Semes
5 }
Elemen dalam Linked List: 7 5 3 11 14 18
Masukkan elemen yang ingin dicari: 11
 Elemen ditemukan pada indeks 3
Masukkan elemen yang ingin dicari: 5
Elemen dalam linked list setelah penghapusan: 7 3 11 14 18
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan 5\GUIDED>
```

UNGUIDED

1. Buatlah ADT Single Linked list sebagai berikut di dalam file “singlelist.h”:

```
Nguided > h singlelist.h > ...
1   // singlelist.h
2   #ifndef SINGLELIST_H
3   #define SINGLELIST_H
4
5   #include <iostream>
6   using namespace std;
7
8   typedef int infotype; // Tipe data untuk info
9   typedef struct ElmList* address; // Pointer ke ElmList
10
11  struct ElmList {
12      infotype info;  // Data di dalam node
13      address next;   // Pointer ke node berikutnya
14  };
15
16  struct List {
17      address First;  // Pointer ke node pertama di dalam list
18  };
19
20  // Prototipe fungsi
    Tabnine | Edit | Test | Explain | Document | Ask
21  void createList(List &L);
    Tabnine | Edit | Test | Explain | Document | Ask
22  address alokasi(infotype x);
    Tabnine | Edit | Test | Explain | Document | Ask
23  void dealokasi(address P);
    Tabnine | Edit | Test | Explain | Document | Ask
24  void insertFirst(List &L, address P);
    Tabnine | Edit | Test | Explain | Document | Ask
25  void printInfo(List L);
26
27  // Deklarasi untuk fungsi baru
    Tabnine | Edit | Test | Explain | Document | Ask
28  address findElm(List L, infotype x);
    Tabnine | Edit | Test | Explain | Document | Ask
29  int sumList(List L);
30
31  #endif
```

Cobalah hasil implementasi ADT pada file “main.cpp”

```
#include "singlelist.h"
#include "singlelist.cpp"

int main() {
   List L;
   address P1, P2, P3, P4, P5 = NULL;

   createList(L);

   P1 = alokasi(2);
   insertFirst(L, P1);

   P2 = alokasi(0);
   insertFirst(L, P2);

   P3 = alokasi(8);
   insertFirst(L, P3);
```

```
    P4 = alokasi(12);
    insertFirst(L, P4);

    P5 = alokasi(9);
    insertFirst(L, P5);

    // Mencetak elemen-elemen dalam list
    printInfo(L); // Output: 9 12 8 0 2

    // Mencari elemen dengan nilai 8
    address found = findElm(L, 8);
    if (found != NULL) {
        cout << found->info << " ditemukan dalam list" << endl; // Output: 8 ditemukan dalam list
    } else {
        cout << "Elemen tidak ditemukan" << endl;
    }

    // Menghitung total elemen dalam list
    int totalSum = sumList(L);
    cout << "Total info dari kelima elemen adalah " << totalSum << endl; // Output: Total info dari kelima elemen adalah 31

    return 0;
}
```

2. Carilah elemen dengan info 8 dengan membuat fungsi baru. fungsi findElm(L : List, x : infotype) : address
3. Carilah elemen dengan info 8 dengan membuat fungsi baru. fungsi findElm(L : List, x : infotype) : address

```
#include "singlelist.h"

// Membuat list dengan mengatur node pertama menjadi NULL
void createList(List &L) {
    L.First = NULL;
}

// Mengalokasikan memori untuk node baru dan mengisi infonya
address alokasi(infotype x) {
    address P = new ElmList;
    P->info = x;
    P->next = NULL;
    return P;
}

// Mendekalokasi memori dari node
```

```
void dealokasi(address P) {
    delete P;
}

// Menyisipkan node di awal list
void insertFirst(List &L, address P) {
    P->next = L.First;
    L.First = P;
}

// Mencetak semua elemen dalam list
void printInfo(List L) {
    address P = L.First;
    while (P != NULL) {
        cout << P->info << " ";
        P = P->next;
    }
    cout << endl;
}

// Fungsi untuk mencari elemen dengan nilai tertentu
address findElm(List L, infotype x) {
    address P = L.First;
    while (P != NULL) {
        if (P->info == x) {
            return P;  // Mengembalikan node jika ditemukan
        }
        P = P->next;
    }
    return NULL;  // Mengembalikan NULL jika tidak ditemukan
}

// Fungsi untuk menghitung total dari semua elemen di dalam list
int sumList(List L) {
    int total = 0;
    address P = L.First;
    while (P != NULL) {
        total += P->info;
        P = P->next;
    }
    return total;
}
```

OutPutnya:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan 5> cd 'd:\Semester5\StrukturData\Prak
d\output'
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan 5\UNguided\output> & .\'main.exe'
9 12 8 0 2
8 ditemukan dalam list
Total info dari kelima elemen adalah 31
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan 5\UNguided\output>
```