

LAPORAN PRAKTIKUM

▪ Identitas Praktikum

Nama MK : Struktur Data
Kode MK : CCK2AAB4
Bobot SKS : 4 SKS
Tempat : L-Program, Gedung DC, lantai 3
Hari, tanggal : Selasa, 22 Oktober 2024
Jam : 12:30-15:30 WIB
Topik praktikum : Modul-5 SINGLE LINKED LIST (BAGIAN KEDUA)

▪ Identitas Mahasiswa

Nama lengkap : Afad Fath Musyarof Halim
NIM : 2211104030
Program Studi : S-1 Software Engineering

▪ Hasil Praktikum

5.1. Searching

Pencarian dalam sebuah list adalah proses iterasi melalui setiap elemen dalam list untuk menemukan elemen tertentu. Proses ini akan berhenti ketika elemen yang dicari ditemukan.

Fungsi yang digunakan untuk Searching di SLL:

```
Node* search(Node* head, int key) {  
    Node* current = head;  
    while (current != nullptr) {  
        if (current->data == key) {  
            return current;  
        }  
        current = current->next;  
    }  
    return nullptr;  
}
```

Penjelasan:

- Node head berisi SLL, int key adalah value yang ingin dicari
- Node Current berisi urutan data saat ini
- Iterasikan selama Current tidak kosong

- Jika data dari Current sesuai dengan key maka kembalian nilai key, jika tidak menuju ke urutan data selanjutnya
- Jika sampai urutan terakhir tidak ditemukan key maka akan mengembalikan nilai nullptr (Tidak ada)

5.2. Latihan

2. Buatlah ADT Single Linked List

- Coding
 - Singlelist.h

```
#ifndef SINGLELIST_H
#define SINGLELIST_H

#include <iostream>

typedef int infotype;
typedef struct ElmList *address;

struct ElmList {
    infotype info;
    address next;
};

struct List {
    address First;
};

void CreateList(List &L);

address alokasi(infotype x);

void dealokasi(address &P);

void printInfo(List &L);

void insertFirst(List &L, address P);

#endif // SINGLELIST_H
```

- Singlelist.cpp

```
#include "singlelist.h"

void CreateList(List &L) {
    L.First = nullptr;
}

address alokasi(infotype x) {
    address P = new ElmList;
    if (P != nullptr) {
        P->info = x;
        P->next = nullptr;
    }
    return P;
}

void dealokasi(address &P) {
    delete P;
    P = nullptr;
}

void printInfo(List &L) {
    address P = L.First;
    while (P != nullptr) {
        std::cout << P->info << " ";
        P = P->next;
    }
    std::cout << std::endl;
}

void insertFirst(List &L, address P) {
    if (P != nullptr) {
        P->next = L.First;
        L.First = P;
    }
}
```

- Main.cpp

```
#include "singelist.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);

    printInfo(L);

    return 0;
}
```

- Output

```
● Afadfath | output
# & .\'main.exe'
9 12 8 0 2
○ Afadfath | output
# []
```

3. Carilah elemen dengan info 8 dengan membuat fungsi baru.

fungsi findElm(L : List, x : infotype) : address

- Coding

- o Tambahan singlelist.h

```
address findElm(List &L, infotype x);
```

- o Tambahan singlelist.cpp

```
address findElm(List &L, infotype x){  
    address P = L.First;  
    while (P != nullptr) {  
        if (P->info == x) {  
            return P;  
        }  
        P = P->next;  
    }  
    return nullptr;  
}
```

- o Tambahan main.cpp

```
address P = findElm(L, 8);  
  
if (P != NULL) {  
    std::cout << P->info << " Ditemukan dalam list" << std::endl;  
} else {  
    std::cout << P->info << " tidak ditemukan" << std::endl;  
}
```

- Output

```
● Afadfath | output  
# & .\'main.exe'  
9 12 8 0 2  
  
8 Ditemukan dalam list  
○ Afadfath | output  
#
```

4. Hitunglah jumlah total info seluruh elemen

- Coding

- o Tambahan singelists.h

```
int totalElm(List &L);
```

- o Tambahan singelists.cpp

```
int totalElm(List &L) {  
    int total = 0;  
    address P = L.First;  
    while (P != nullptr) {  
        total += P->info;  
        P = P->next;  
    }  
    return total;  
}
```

- o Tambahan main.cpp

```
std::cout << "Total info dari kelima elemen adalah "  
    << totalElm(L)  
    << std::endl;
```

- Output

```
● Afadfath | output  
# & .\'main.exe'  
9 12 8 0 2  
  
8 Ditemukan dalam list  
  
Total info dari kelima elemen adalah 31  
○ Afadfath | output  
#
```