

LAPORAN PRAKTIKUM
STRUKTUR DATA
SINGLE LINKED LIST BAGIAN 1



Nama :

Muhammad Mahrus Ali (2311104006)

Dosen :

Yudha Islami Sulistya,S.Kom.,M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

A. Soal Tugas Pendahuluan

1. Membuat Deklarasi Tipe List

Buat file list.h dan ketik sintak berikut

```
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info
using namespace std;
typedef int infotype;
typedef struct elmlist *address;

struct elmlist {
    infotype info;
    address next;
};

struct List{
    address first;
};
```

Buat file list.cpp dan ketik sintak berikut

```
#include <iostream>
#include "list.h"
using namespace std;
```

2. Membuat list kosong, yaitu procedure createList

Tambahkan pada list.h primitive dari procedure createList

```
void createList(List &L);
```

Tambahkan pada list.cpp implementasi dari procedure createList, sintak c++ sbb:

```
void createList(List &L) {
    /** this procedure will initialize
        the list L*/
    first(L) = NULL;
}
```

3. Setelah list sudah ada, selanjutnya buatlah element dengan menggunakan fungsi allocate

Tambahkan pada list.h primitive dari fungsi allocate

```
address allocate(infotype x);
```

Tambahkan pada list.cpp implementasi dari fungsi allocate, sintak c++ sbb:

```
address allocate(infotype x) {
    address p = new elmlist;
    info(p) = x;
    next(p) = NULL;

    return p;
}
```

4. Setelah List dan elemen sudah ada, maka selanjutnya elemen tersebut harus diinsert ke List agar bisa menjadi elemen list.

Proses insert dapat menggunakan procedure Insert First, procedure Insert Last, atau procedure insert After. Pada Tugas Pendahuluan kali ini, akan dicontohkan menggunakan insert first.

Tambahkan pada list.h primitif procedure insertFirst

```
void insertFirst(List &L, address P);
```

Tambahkan list.cpp implementasi dari procedure Insert First sesuai sintak berikut

```
void insertFirst(List &L, address P){
}    /** TODO: Insert the new element pointed
    by P to the first of list L*/
    // YOUR CODES HERE
    //-----
    next(P) = first(L);
    first(L) = P;
    //-----
}
```

5. Berikan Setelah proses insert elemen, maka agar bisa mengetahui apakah elemen berhasil diinsertkan, maka kita perlu menampilkan isi list.

Tambahkan pada list.h primitif procedure printInfo

```
void printInfo (List L);
```

Tambahkan list.cpp implementasi dari procedure printInfo sesuai sintak berikut:

```
void printInfo(List L){
}    /** this procedure will output
    the info of each element
    in list L*/

    address p = first(L);
    while (p != NULL){
        cout << info(p) << ", ";
        p = next(p);
    }
    cout << endl;
}
```

6. Sekarang, setelah ADT List sudah terisi dengan beberapa fungsi Procedur di atas, maka mari buat sebuah List berisi 3 elemen yang berisi 3 digit nim terakhir Anda di main.cpp Adapun gambaran isi dari main.cpp nya adalah sbb :

```

1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
5
6  int main()
7  {
8      // 1. Panggilah create list
9
10     // 2. Buat sintak memanggil angka pertama yang ingin dimasukkan user ke list
11
12     // 3. Panggil fungsi allocate agar data tersebut diisikan elemen
13
14     // 4. Panggil procedure insert first yang telah dibuat
15
16     // 5. Panggil procedure show info untuk mengecek apakah angka tersebut berhasil menjadi elemen di List.
17
18     // 6. buat kembali sintak no 2 s/d no 5 untuk data angka kedua dari user
19
20     // 7. buat kembali sintak no 2 s/d no 5 untuk data angka ketiga dari user
21
22
23     return 0;
24 }
25
26

```

B. Latihan Unguided

1. Buat program dengan output seperti berikut

SESI HAVE FUN. Rekan-rekan dapat mencoba hal di bawah ini agar memudahkan saat praktikum:

- Tambahkan procedure insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp
- Tambahkan Function searchInfo pada list.h dan list.cpp
- Ubah main.cpp agar proses insert N data tidak satu persatu, tapi sesuai dengan jumlah digit NIM yaitu 10 data (clue : gunakan looping). Dan NIM yang diinput, saat di show tidak boleh terurut terbalik (clue : gunakan insert Last) Tampilan

(underscore adalah inputan user):

Masukkan NIM perdigit

Digit 1 : 1

Digit 2 : 1

Digit 3 : 3

Digit 4 : 1

Digit 5 : 9

Digit 6 : 6

Digit 7 : 4

Digit 8 : 7

Digit 9 : 4

Digit 10 : 2

Isi list : 1131964742

Inputan :

File List.cpp

```

#include <iostream>
#include "list.h"

using namespace std;

void createlist(List &L) {
    first(L) = NULL;
}

address allocate(infotype x) {
    address p = new elmlist;
    info(p) = x;
    next(p) = NULL;

    return p;
}

void insertFirst(List &L, address P) {
    next(P) = first(L);
    first(L) = P;
}

void insertLast(List &L, address P) {
    if (first(L) == NULL) {
        insertFirst(L, P);
    } else {
        address Q = first(L);

        while (next(Q) != NULL) {
            Q = next(Q);
        }

        next(Q) = P;
    }
}

void insertAfter(List &L, address P, int Q) {
    address Prec = first(L);

    while ((next(Prec) != NULL) && (info(Prec) != Q)) {
        Prec = next(Prec);
    }

    if (next(Prec) != NULL) {
        next(P) = next(Prec);
        next(Prec) = P;
    }
}

void deleteFirst(List &L) {
    address P = first(L);
    first(L) = next(P);
    next(P) = NULL;
}

void deleteLast(List &L) {
    address Q = first(L);

    while (next(next(Q)) != NULL) {
        Q = next(Q);
    }

    address P = next(Q);
    next(Q) = NULL;
}

void deleteAfter(List &L, int Q) {
    address Prec = first(L);

    while ((next(Prec) != NULL) && (info(Prec) != Q)) {
        Prec = next(Prec);
    }

    address P = next(Prec);
    next(Prec) = next(P);
    next(P) = NULL;
    delete P;
}

void printInfo(List L) {
    address p = first(L);

    cout << "Isi list : ";

    while (p != NULL) {
        cout << info(p);
        p = next(p);
    }
    cout << endl;
}

int searchInfo(List &L, int P) {
    address Q = first(L);
    address temp = NULL;

    while (Q != NULL) {
        if (info(Q) == P) {
            if (temp == NULL) {
                cout << "Nilai " << P << "pada urutan pertama" << endl;
            } else {
                cout << "Nilai " << P << "setelah nilai " << info(temp) << endl;
            }
            return true;
        }
        temp = Q;
        Q = next(Q);
    }

    cout << "Nilai " << P << " tidak ditemukan!" << endl;
    return false;
}

```

File List.h

```
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info

using namespace std;

typedef int infotype;
typedef struct elmList *address;

struct elmList {
    infotype info;
    address next;
};

struct List {
    address first;
};

void createList(List &L);

address allocate(infotype x);

void insertFirst(List &L, address P);
void insertLast(List &L, address P);
void insertAfter(List &L, address P, address Prec);

void deleteFirst(List &L);
void deleteLast(List &L);
void deleteAfter(List &L, address Prec);

void printInfo(List L);

int searchInfo(List &L, int P);
```

File Main.cpp

```
#include <iostream>
#include "list.cpp"

using namespace std;

int main() {
    List head;

    createList(head);

    cout << "Masukkan NIM perdigit" << endl;

    for (int i = 0; i < 10; i++) {
        int number;

        cout << "Digit " << i+1 << " : ";
        cin >> number;

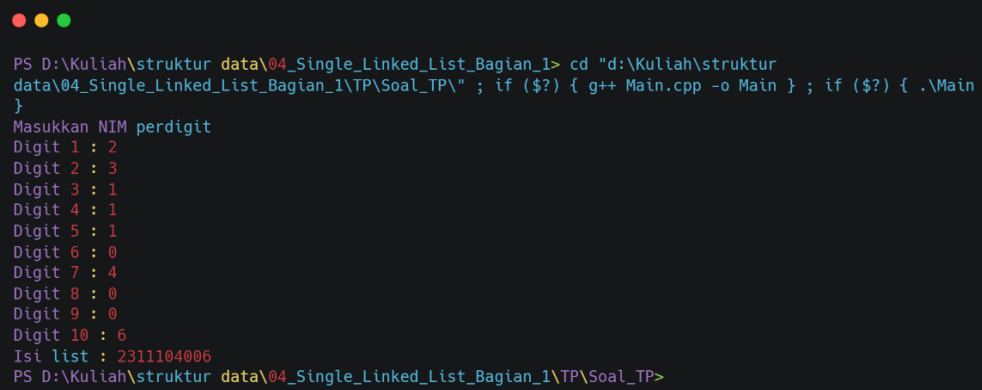
        address elm = allocate(number);

        insertLast(head, elm);
    }

    printInfo(head);

    return 0;
}
```

Outputnya adalah :



```
PS D:\Kuliah\struktur data\04_Single_Linked_List_Bagian_1> cd "d:\Kuliah\struktur
data\04_Single_Linked_List_Bagian_1\TP\Soal_TP\" ; if ($?) { g++ Main.cpp -o Main } ; if ($?) { .\Main
}
Masukkan NIM perdigit
Digit 1 : 2
Digit 2 : 3
Digit 3 : 1
Digit 4 : 1
Digit 5 : 1
Digit 6 : 0
Digit 7 : 4
Digit 8 : 0
Digit 9 : 0
Digit 10 : 6
Isi list : 2311104006
PS D:\Kuliah\struktur data\04_Single_Linked_List_Bagian_1\TP\Soal_TP>
```