

PRAKTIKUM STRUKTUR DATA

TUGAS PENDAHULUAN 04

Single Linked List



Nama :

Raihan Sastra Wibyanto (2311104020)

Dosen :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

B. Soal Praktek

Ikuti langkah-langkah berikut untuk mengerjakan TP modul 4 Materi Single Linked List :

1. Membuat deklarasi tipe List

Buat file list.h dan ketik sintak sesuai gambar berikut.

```
1  #include <iostream>
2  #define first(L) L.first
3  #define next(P) P->next
4  #define info(P) P->info
5  using namespace std;
6  typedef int infotype;
7  typedef struct elmlist *address;
8
9  // Membuat deklarasi tipe list
10 struct elmlist
11 {
12     infotype info;
13     address next;
14 }
15 };
16
17 struct List
18 {
19     address first;
20 };
```

Buat file list.cpp dan ketik sintak berikut

```
1  #include <iostream>
2  #include "List.h"
3  using namespace std;
```

2. Membuat list kosong, yaitu procedure createList.

Tambahkan pada list.h primitif dari procedure createList

```
22 // Membuat procedure createList
23 void createList(List &L);
```

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

```

6 // Implementasi dari procedure createList
7 void createList(List &L){
8     /* this procedure will initialize the list L */
9     first(L) = NULL;
10 }

```

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

Tambahkan pada list.h primitif dari fungsi allocate

```

25 // Membuat elemen dengan menggunakan fungsi allocate
26 address allocate(infotype x);

```

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

```

12 // Implementasi dari fungsi allocate
13 address allocate(infotype x){
14
15     address P = new elmlist;
16     info(P) = x;
17     next(P) = NULL;
18
19     return P;
20 }

```

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

```

28 // Elemen harus diinsert ke List
29 void insertFirst(List &L, address P);

```

Tambahkan pada list.cpp implementasi dari procedure insertFirst sesuai sintak berikut :

```

22 // Implementasi dari procedure insertFirst
23 void insertFirst(List &L, address P){
24     /* TODO: Insert the new element pointed by F to the first of list L */
25     // YOUR CODES HERE
26     //-----
27     next(P) = first(L);
28     first(L) = P;
29     //-----
30 }

```

5. 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

```

31 // Menampilkan isi list
32 void printInfo(List L);

```

Tambahkan pada list.cpp implementasi dari proc printInfo, sintak C++ sebagai berikut :

```

32 // Implementasi dari proc printInfo
33 void printInfo(List L){
34     /* this procedure will output the info of each element in list L */
35     address P = first(L);
36     while (P != NULL){
37         cout << info(P) << ", ";
38         P = next(P);
39     }
40     cout << endl;
41 }

```

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  int main()
6  {
7      List L;
8
9      // 1. Panggilah Create List
10     createList(L);
11
12     // 2. Buat Sintak menanyakan angka pertama yang ingin diinputkan user ke list
13     int angka ;
14     cout << "Masukkan angka pertama: ";
15     cin >> angka;
16
17     // 3. Panggil fungsi allocate agar data tersebut dijadikan elemen
18     address P = allocate(angka);
19
20     // 4. Panggil procedure insert first yang telah dibuat
21     insertFirst(L, P);
22
23     // 5. panggil procedure show info untuk mengecek apakah angka tersebut berhasil menjadi
24     cout << "Isi list setelah insert pertama: ";
25     printInfo(L);
26
27     // 6. buat kembali sintak no 2 s/d no 5 untuk data  angka kedua dari user
28     cout << "Masukkan angka kedua: ";
29     cin >> angka;
30     P = allocate(angka);
31     insertFirst(L, P);
32     cout << "Isi list setelah insert kedua: ";
33     printInfo(L);
34
35     // 7. buat kembali sintak no 2 s/d no 5 untuk data angka ketiga dari user
36     cout << "Masukkan angka kedua: ";
37     cin >> angka;
38     P = allocate(angka);
39     insertFirst(L, P);
40     cout << "Isi list setelah insert ketiga: ";
41     printInfo(L);
42
43     return 0;
44 }

```

Hasil outputnya:

```

Raihan Sastra W@MSI MINGW64 ~/Documents/Institut Teknologi Telkom Purwokerto/Semester 3/Struktur Data/S
TD_Raihan_Sastra_Wibyanto_2311104020/04_Single_Linked_List_Bagian1/TP (main)
$ ./main
Masukkan angka pertama: 0
Isi list setelah insert pertama: 0,
Masukkan angka kedua: 2
Isi list setelah insert kedua: 2, 0,
Masukkan angka kedua: 0
Isi list setelah insert ketiga: 0, 2, 0,

```

Karena angka pertama dan terakhir 0 jadi sama tetapi jika diganti jadi 123 maka akan terbalik jadi 321.

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

File main.cpp

```
1  #include <iostream>
2  #include "List.h"
3  using namespace std;
4
5  int main()
6  {
7      List L;
8
9      // 1. Panggilah Create List
10     createList(L);
11
12     cout << ("\n");
13     /* SOAL NO 7 */
14     int nim, digit;
15     cout << "Masukkan NIM per digit:" << endl;
16
17     for (int i = 1; i <= 10; i++) {
18         cout << "Digit " << i << ": ";
19         cin >> digit;
20
21         // Buat node baru dan masukkan di akhir list
22         address P = allocate(digit);
23         insertLast(L, P); // Gunakan insertLast agar urutan tidak terbalik
24     }
25
26     // Cetak isi list (NIM)
27     cout << "Isi list: ";
28     printInfo(L); // Menampilkan NIM yang diinput
29     cout << endl;
30
31     return 0;
32 }
```

File List.h

```

1  #include <iostream>
2  #define first(L) L.first
3  #define next(P) P->next
4  #define info(P) P->info
5  using namespace std;
6  typedef int infotype;
7  typedef struct elmlist *address;
8
9  // Membuat deklarasi tipe list
10 struct elmlist
11 {
12     infotype info;
13     address next;
14 }
15 };
16
17 struct List
18 {
19     address first;
20 };
21
22 // Membuat procedure createList
23 void createList(List &L);
24
25 // Membuat elemen dengan menggunakan fungsi allocate
26 address allocate(infotype x);
27
28 // Elemen harus diinsert ke List
29 void insertFirst(List &L, address P);
30
31 // Menampilkan isi list
32 void printInfo(List L);
33
34 // tambah procedure iL,iA,dL,dA
35 void createList(List &L);
36 address allocate(infotype x);
37 void insertFirst(List &L, address P);
38 void insertLast(List &L, address P); // New function
39 void insertAfter(address Prec, address P);
40 void deleteFirst(List &L, address &P);
41 void deleteLast(List &L, address &P); // New function
42 void deleteAfter(address Prec, address &P);
43 address searchInfo(List L, infotype x); // New function
44 void printInfo(List L);

```

File List.cpp

```

43  /* SOAL NO 7 */
44  // Insert last: Insert node at the end of the list
45  void insertLast(List &L, address P) {
46      if (first(L) == NULL) {
47          first(L) = P;
48      } else {
49          address temp = first(L);
50          while (next(temp) != NULL) {
51              temp = next(temp);
52          }
53          next(temp) = P;
54      }
55  }
56
57  // Search info: Function to search for an element in the list
58  address searchInfo(List L, infotype x) {
59      address P = first(L);
60      while (P != NULL) {
61          if (info(P) == x) {
62              return P;
63          }
64          P = next(P);
65      }
66      return NULL; // Return NULL if not found
67  }
68
69  // Delete last: Delete the last node from the list
70  void deleteLast(List &L, address &P) {
71      if (first(L) == NULL) {
72          cout << "List is empty" << endl;
73      } else if (next(first(L)) == NULL) {
74          P = first(L);
75          first(L) = NULL;
76      } else {
77          address temp = first(L);
78          while (next(next(temp)) != NULL) {
79              temp = next(temp);
80          }
81          P = next(temp);
82          next(temp) = NULL;
83      }
84  }

```

Output:


```
Raihan Sastra W@MSI MINGW64 ~/Documents/Institut Teknologi Telkom Purwokerto/Semester 3/Struktur Data/ST
TD_Raihan_Sastra_Wibyanto_2311104020/04_Single_Linked_List_Bagian1/TP (main)
$ ./main

Masukkan NIM per digit:
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: 2
Digit 10: 0
Isi list: 2, 3, 1, 1, 1, 0, 4, 0, 2, 0,
```