

LAPORAN PRAKTIKUM
MODUL 5
SINGLE LINKED LIST (BAGIAN KEDUA)



Nama :

Muhammad Rifqi Al Baqi Ananta (2311104005)

Dosen :

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

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

I. TP

1. Soal 1

Code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int info;
6     struct Node* next;
7 };
8 typedef struct Node* address;
9
10 typedef struct {
11     address head;
12 } List;
13
14 address createNode_2311104005(int nilai) {
15     address nodeBaru = (address)malloc(sizeof(struct Node));
16     if (nodeBaru != NULL) {
17         nodeBaru->info = nilai;
18         nodeBaru->next = NULL;
19     }
20     return nodeBaru;
21 }
22
23 void insertFirst_2311104005(List *L, int nilai) {
24     address nodeBaru = createNode_2311104005(nilai);
25     if (nodeBaru != NULL) {
26         nodeBaru->next = L->head;
27         L->head = nodeBaru;
28     }
29 }
30
31 int searchElement_2311104005(List L, int i) {
32     address current = L.head;
33     int posisi = 1;
34
35     while (current != NULL && current->info != i) {
36         current = current->next;
37         posisi++;
38     }
39
40     if (current != NULL) {
41         printf("Elemen ditemukan pada alamat %p, urutan ke-%d\n", current, posisi);
42         return posisi;
43     } else {
44         printf("Elemen tidak ditemukan dalam list\n");
45         return -1;
46     }
47 }
48
49 void printList_2311104005(List L) {
50     address current = L.head;
51     printf("List: ");
52     while (current != NULL) {
53         printf("%d ", current->info);
54         current = current->next;
55     }
56     printf("\n");
57 }
58
59 void freeList_2311104005(List *L) {
60     address current = L->head;
61     while (current != NULL) {
62         address temp = current;
63         current = current->next;
64         free(temp);
65     }
66     L->head = NULL;
67 }
68
69 int main() {
70     List L;
71     L.head = NULL;
72     int nilaiCari;
73
74     printf("=== Program Pencarian Elemen dalam Single Linked List ===\n");
75     printf("Oleh: Nanta (2311104005)\n\n");
76
77     // Memasukkan 6 elemen
78     printf("Masukkan 6 bilangan bulat:\n");
79     for(int i = 0; i < 6; i++) {
80         int nilai;
81         printf("Masukkan angka ke-%d: ", i+1);
82         scanf("%d", &nilai);
83         insertFirst_2311104005(&L, nilai);
84     }
85
86     printf("\nIsi list saat ini: ");
87     printList_2311104005(L);
88
89     printf("\nMasukkan nilai yang ingin dicari: ");
90     scanf("%d", &nilaiCari);
91     searchElement_2311104005(L, nilaiCari);
92
93     freeList_2311104005(&L);
94
95     return 0;
96 }
```

Output:

```
Masukkan 6 bilangan bulat:
Masukkan angka ke-1: 10
Masukkan angka ke-2: 13
Masukkan angka ke-3: 15
Masukkan angka ke-4: 17
Masukkan angka ke-5: 19
Masukkan angka ke-6: 21

Isi list saat ini: List: 21 19 17 15 13 10

Masukkan nilai yang ingin dicari: 13
Elemen ditemukan pada alamat 007915D8, urutan ke-5
PS D:\TUGAS ITTP\SEMESTER 3\Praktikum Struktur Data\Pertemuan 5\TP_5\output>
```

2. Soal 2

Code:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Node {
5      int info;
6      struct Node* next;
7  };
8  typedef struct Node* address;
9
10 typedef struct {
11     address head;
12 } List;
13
14 address createNode_2311104005(int nilai) {
15     address nodeBaru = (address)malloc(sizeof(struct Node));
16     if (nodeBaru != NULL) {
17         nodeBaru->info = nilai;
18         nodeBaru->next = NULL;
19     }
20     return nodeBaru;
21 }
22
23 void insertFirst_2311104005(List *L, int nilai) {
24     address nodeBaru = createNode_2311104005(nilai);
25     if (nodeBaru != NULL) {
26         nodeBaru->next = L->head;
27         L->head = nodeBaru;
28     }
29 }
30
31 void bubbleSort_2311104005(List *L) {
32     int ditukar;
33     address current;
34     address akhir = NULL;
35
36     if (L->head == NULL)
37         return;
38
39     do {
40         ditukar = 0;
41         current = L->head;
42
43         while (current->next != akhir) {
44             if (current->info > current->next->info) {
45                 int temp = current->info;
46                 current->info = current->next->info;
47                 current->next->info = temp;
48                 ditukar = 1;
49             }
50             current = current->next;
51         }
52         akhir = current;
53     } while (ditukar);
54 }
55
56 void printList_2311104005(List L) {
57     address current = L.head;
58     printf("List: ");
59     while (current != NULL) {
60         printf("%d ", current->info);
61         current = current->next;
62     }
63     printf("\n");
64 }
65
66 void freeList_2311104005(List *L) {
67     address current = L->head;
68     while (current != NULL) {
69         address temp = current;
70         current = current->next;
71         free(temp);
72     }
73     L->head = NULL;
74 }
75
76 int main() {
77     List L;
78     L.head = NULL;
79
80     printf("Masukkan 5 bilangan bulat:\n");
81     for(int i = 0; i < 5; i++) {
82         int nilai;
83         printf("Masukkan angka ke-%d: ", i+1);
84         scanf("%d", &nilai);
85         insertFirst_2311104005(&L, nilai);
86     }
87
88     printf("\nSebelum diurutkan: ");
89     printList_2311104005(L);
90
91     bubbleSort_2311104005(&L);
92
93     printf("Setelah diurutkan: ");
94     printList_2311104005(L);
95
96     freeList_2311104005(&L);
97
98     return 0;
99 }
100

```

Output:

```
Masukkan 5 bilangan bulat:  
Masukkan angka ke-1: 10  
Masukkan angka ke-2: 13  
Masukkan angka ke-3: 14  
Masukkan angka ke-4: 16  
Masukkan angka ke-5: 20  
  
Sebelum diurutkan: List: 20 16 14 13 10  
Setelah diurutkan: List: 10 13 14 16 20  
PS D:\TUGAS ITTP\SEMESTER 3\Praktikum Struktur Data\Pertemuan 5\TP_5\output>
```

3. Soal 3

Code:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Node {
5      int info;
6      struct Node* next;
7  };
8  typedef struct Node* address;
9
10 typedef struct {
11     address head;
12 } List;
13
14 address createNode_2311104005(int nilai) {
15     address nodeBaru = (address)malloc(sizeof(struct Node));
16     if (nodeBaru != NULL) {
17         nodeBaru->info = nilai;
18         nodeBaru->next = NULL;
19     }
20     return nodeBaru;
21 }
22
23 void insertSorted_2311104005(List *L, address P) {
24     address Q = L->head;
25     address Prev = NULL;
26     int found = 0;
27
28     while (Q != NULL && !found) {
29         if (Q->info < P->info) {
30             Prev = Q;
31             Q = Q->next;
32         } else {
33             found = 1;
34         }
35     }
36
37     if (Q == L->head) {
38         P->next = L->head;
39         L->head = P;
40     }
41     else if (Q == NULL) {
42         Prev->next = P;
43         P->next = NULL;
44     }
45     else {
46         P->next = Q;
47         Prev->next = P;
48     }
49 }
50
51 void printList_2311104005(List L) {
52     address current = L.head;
53     printf("List: ");
54     while (current != NULL) {
55         printf("%d ", current->info);
56         current = current->next;
57     }
58     printf("\n");
59 }
60
61 void freeList_2311104005(List *L) {
62     address current = L->head;
63     while (current != NULL) {
64         address temp = current;
65         current = current->next;
66         free(temp);
67     }
68     L->head = NULL;
69 }
70
71 int main() {
72     List L;
73     L.head = NULL;
74
75     printf("Masukkan 4 bilangan bulat secara terurut (dari kecil ke besar):\n");
76     for(int i = 0; i < 4; i++) {
77         int nilai;
78         printf("Masukkan angka ke-%d: ", i+1);
79         scanf("%d", &nilai);
80         address nodeBaru = createNode_2311104005(nilai);
81         insertSorted_2311104005(&L, nodeBaru);
82     }
83
84     printf("\nList saat ini: ");
85     printList_2311104005(L);
86
87     printf("\nMasukkan nilai baru yang akan disisipkan: ");
88     int nilaiBaru;
89     scanf("%d", &nilaiBaru);
90     address nodeBaru = createNode_2311104005(nilaiBaru);
91     insertSorted_2311104005(&L, nodeBaru);
92
93     printf("Setelah penyisipan: ");
94     printList_2311104005(L);
95
96     freeList_2311104005(&L);
97
98     return 0;
99 }
100 }

```

Output:

```
Masukkan 4 bilangan bulat secara terurut (dari kecil ke besar):  
Masukkan angka ke-1: 10  
Masukkan angka ke-2: 14  
Masukkan angka ke-3: 16  
Masukkan angka ke-4: 18  
  
List saat ini: List: 10 14 16 18  
  
Masukkan nilai baru yang akan disisipkan: 13  
Setelah penyisipan: List: 10 13 14 16 18  
PS D:\TUGAS ITTP\SEMESTER 3\Praktikum Struktur Data\Pertemuan 5\TP_5\output>
```

II. PENJELASAN

1. Soal 1 - Program Pencarian Elemen dalam Single Linked List

- Program ini membuat linked list dan mencari posisi elemen tertentu

- Fitur utama:

- Menerima input 6 bilangan
- Memasukkan bilangan ke awal list (insertFirst)
- Mencari elemen dan menampilkan posisi serta alamatnya
- Jika elemen ditemukan, mengembalikan posisinya, jika tidak mengembalikan -1

2. Soal 2 - Program Pengurutan Linked List

- Program ini mengurutkan linked list menggunakan bubble sort

- Fitur utama:

- Menerima input 5 bilangan
- Memasukkan bilangan ke awal list
- Mengurutkan list dari kecil ke besar dengan bubble sort
- Menampilkan list sebelum dan sesudah pengurutan

3. Soal 3 - Program Penyisipan Terurut dalam Linked List

- Program ini menyisipkan elemen baru ke dalam linked list terurut

- Fitur utama:

- Menerima input 4 bilangan secara terurut
- Menyisipkan bilangan baru dengan mempertahankan urutan
- Menggunakan insertSorted untuk menempatkan elemen baru di posisi yang tepat
- Menampilkan list sebelum dan sesudah penyisipan