

IK141 Struktur Data

Struktur Data

Double Linked List



Di Susun Oleh :

Wendi Kardan, 2100016

PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER

FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PENDIDIKAN INDONESIA

20 Maret 2022

1. Implementasi dan Hasil

Implementasi dan Hasil

A. Studi Kasus

1. Data yang disimpan adalah data ID, Nama, Gender dan Usia
2. Fungsi/prosedur yang harus dibuat dengan key tertentu:
 - a. Fungsi Create Element baru
 - b. Key 1 : Fungsi Menambah elemen di awal
 - c. Key 2 : Prosedur Mencetak data dalam element tertentu, misalnya cetak data dari Jemaah dengan ID tertentu
 - d. Key 3: Prosedur addAfter, yaitu sebuah prosedur yang digunakan untuk menambahkan elemen setelah ID tertentu
 - e. Key 4: Prosedur addBefore, yaitu sebuah prosedur yang digunakan untuk menambahkan elemen sebelum ID tertentu
 - f. Key 5 : Prosedur addAkhir, yaitu menambahkan element diakhir
 - g. Key 6 : Prosedur delete element dengan ID tertentu
 - h. Key 7 : Prosedur mencetak data semua element dalam linked list

Program akan selalu meminta untuk diinputkan key 1-7 sampai key yang diinputkan adalah 0. 4. Jika key diinputkan 0, maka program akan mencetak (jika ada fungsi cetak) dan keluar

B. Souces Code

1. Import library yang akan digunakan serta melakukan inisialisasi struct node yang di dalamnya berisikan data (id, nama, gender, usia), next address, serta prev address. Setelah membuat struct buat sebuah variabel yang bertipe data struct tersebut dengan initial value NULL.

```
C Praktikum5.c
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4
5  // Nama : Wendi Kardan
6  // NIM : 2100016
7  // Kelas : Pendidikan Ilmu Komputer - A
8  // Double linked list (Praktikum 5)
9
10
11 // inisialisasi double linked list menggunakan struct
12 struct node{
13     int id;
14     char nama[20];
15     char gender[20];
16     int usia;
17     struct node *next;
18     struct node *prev;
19 };
20
21 // deklarasi variabel untuk linked list dengan initial value NULL
22 struct node* start = NULL;
```

2. Fungsi `createElement` untuk membuat nodes baru yang nantinya akan ditambahkan kedalam double linked list dan data yang dimasukkannya merupakan input dari user.

```
//function untuk membuat linkedlist yang baru
struct node* createElement(){
    struct node *new_node;
    new_node = (struct node*)malloc(sizeof(struct node));
    int id, usia;
    char nama[20], gender[20];
    scanf("%d",&id);
    scanf("%s", nama);
    scanf("%s", gender);
    scanf("%d",&usia);
    new_node->id = id;
    new_node->usia = usia;
    strcpy(new_node->nama,nama);
    strcpy(new_node->gender,gender);
    new_node->next = NULL;
    new_node->prev = NULL;
    return new_node;
}
```

3. Function untuk menambahkan node baru di awal linked list. Di dalamnya akan dicek apabila start masih null maka ia akan membuat element baru, apabila tidak dia akan menyisipkan data diawal.

```
44 // menambahkan nodes baru di awal linked list
45 struct node* addawal(struct node *start){
46     struct node *new_node;
47     if(start == NULL){
48         new_node = createElement();
49         start = new_node;
50     }else{
51         new_node = createElement();
52         new_node->prev = NULL;
53         new_node->next = start;
54         start->prev = new_node;
55         start = new_node;
56     }
57     return start;
58 }
```

4. Function `display` untuk menampilkan seluruh nodes yang ada di dalam linked list.

```
60 // function untuk menampilkan seluruh data di seluruh linked list
61 struct node* display(struct node *start){
62     struct node *ptr;
63     ptr = start;
64     while(ptr != NULL){
65         printf("%d\n%s\n%s\n%d\n\n",ptr->id, ptr->nama, ptr->gender, ptr->usia);
66         ptr = ptr->next;
67     }
68     return start;
69 }
70
```

5. Function untuk menambahkan data setelah id tertentu

```
71 //function untuk menambahkan data nodes setelah id tertentu
72 struct node *addAfter(struct node *start){
73     struct node *new_node, *ptr;
74     int askedId;
75     scanf("%d", &askedId);
76     ptr = start;
77     while(ptr->id != askedId){
78         ptr = ptr->next;
79     }
80     new_node = createElement();
81     if(ptr->next == NULL){
82         ptr->next = new_node;
83         new_node->prev = ptr;
84         new_node->next = NULL;
85     }else{
86         new_node->prev = ptr;
87         new_node->next = ptr->next;
88         ptr->next->prev = new_node;
89         ptr->next = new_node;
90     }
91     return start;
92 }
93 }
```

6. Menambahkan node baru sebelum id yang diberikan

```
95 // menambahkan nodes baru sebelum id tertentu
96 struct node *addBefore(struct node *start){
97     struct node *new_node, *ptr;
98     int askedId;
99     scanf("%d", &askedId);
100     ptr = start;
101     while(ptr->id != askedId){
102         ptr = ptr->next;
103     }
104     new_node = createElement();
105     if(ptr == start){
106         new_node->prev = NULL;
107         new_node->next = start;
108         start->prev = new_node;
109         start = new_node;
110     }else{
111         new_node->next = ptr;
112         new_node->prev = ptr->prev;
113         ptr->prev->next = new_node;
114         ptr->prev = new_node;
115     }
116     return start;
117 }
```

7. Menambahkan nodes baru di akhir double linked list

```

119 // menambahkan nodes baru di akhir linked list tersebut
120 struct node *addAkhir(struct node *start){
121     struct node *new_node, *ptr;
122     ptr = start;
123     while(ptr->next != NULL){
124         ptr = ptr->next;
125     }
126     new_node = createElement();
127     ptr->next = new_node;
128     new_node->prev = ptr;
129     new_node->next = NULL;
130     return start;
131 }

```

8. Menghapus nodes dengan id tertentu di dalam double linked list, (apabila nodenya diawal maka deleteAwal, apabila diakhir deleteAkhir, apabila ditengah maka deleteTengah)

```

133 // menghapus data di linked list tersebut berdasarkan id yang dimasukan oleh user
134 struct node *delete(struct node *start){
135     struct node *ptr;
136     ptr = start;
137     int askedId;
138     scanf("%d", &askedId);
139     while(ptr->id != askedId){
140         ptr = ptr->next;
141     }
142     if(ptr == start){
143         start = start->next;
144         start->prev = NULL;
145         free(ptr);
146     }else if(ptr->next == NULL){
147         ptr->prev->next = NULL;
148         free(ptr);
149     }
150     else{
151         ptr->prev->next = ptr->next;
152         ptr->next->prev = ptr->prev;
153         free(ptr);
154     }
155     return start;
156 }
157

```

9. Function untuk menampilkan nodes dalam linked list berdasarkan id tertentu.

```

// menampilkan data berdasarkan id dari nodes yang di inputkan
struct node *displayDataId(struct node *start){
    struct node *ptr;
    ptr = start;
    int askedId;
    scanf("%d", &askedId);
    while(ptr->id != askedId){
        ptr = ptr->next;
    }
    printf("\n%d\n%s\n%s\n%d\n", ptr->id, ptr->nama, ptr->gender, ptr->usia);
    return start;
}

```

10. Function utama yang akan dieksekusi pertama kali oleh program yang di dalamnya terdapat menu menggunakan switch case, dimana masing masing case akan memanggil function tersebut.

```
172 // function utama
173 int main(){
174     int option; //variabel untuk opsi user
175     do{
176         scanf("%d",&option); //
177         switch(option){
178             //addawal
179             case 1: //menambah data di awal
180                 start = addawal(start);
181                 break;
182             case 2: //menampilkan data berdasarkan id tertentu
183                 start = displayDataId(start);
184                 break;
185             case 3: //menambahkan data setelah id tertentu
186                 start = addAfter(start);
187                 break;
188             case 4: //menambahkan data sebelum id tertentu
189                 start = addBefore(start);
190                 break;
191             case 5: //menambahkan data di akhir linked list
192                 start = addAkhir(start);
193                 break;
194             case 6: //menghapus data berdasarkan id tertentu
195                 start = delete(start);
196                 break;
197             case 7: //menampilkan seluruh data yang ada di dalam linked list
198                 if(start == NULL){
199                     case 7: //menampilkan seluruh data yang ada di dalam linked list
200                     if(start == NULL){
201                         printf("Tidak ada\n");
202                     }else{
203                         start = display(start);
204                     }
205                     break;
206                 case 0: // menampilkan data (di cek terlebih dahulu)
207                 if(start == NULL){
208                     printf("Tidak ada");
209                 }else{
210                     start = display(start);
211                 }
212                 break;
213             }
214         }while(option != 0); //apabila input == 0 maka akan outdari looping dan program selesai
215     }
216     return 1;
217 }
```

PROSES PENGUJIAN

A. Case 1

```
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 6\Praktikum 5> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 6\Praktikum 5\" ; if ($?) { gcc Praktikum5.c -o Praktikum5 } ; if ($?) { .\Praktikum5 }
0
Tidak ada
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 6\Praktikum 5> █
```

B. Case2

```
UKTUR DATA\Pertemuan 6\Praktikum 5\" ; if ($?) { gcc Praktikum5.c -o Praktikum5 } ; i
f ($?) { .\Praktikum5 }
1
107
Firman
Laki-laki
55
1
106
Julia
Perempuan
45
5
108
Hilman
Laki-laki
55
6
107
7
106
Julia
Perempuan
45

108
Hilman
Laki-laki
55

0
106
Julia
Perempuan
45

108
Hilman
```

C. Case3

```
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Pertemuan 6\Praktikum 5> cd "d:\UPI\SEMESTER 2\STR
UKTUR DATA\Pertemuan 6\Praktikum 5\" ; if ($?) { gcc Praktikum5.c -o Praktikum5 } ; i
f ($?) { .\Praktikum5 }
1
107
Firman
Laki-laki
55
5
108
Hilman
Laki-laki
55
2 108

108
Hilman
Laki-laki
55
[]
```

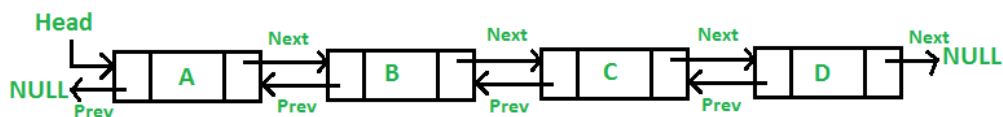
D. Case4

```
f ($?) { .\Praktikum5 }  
1  
107  
Firman  
Laki-laki  
55  
5  
108  
Hilman  
Laki-laki  
55  
2 108  
  
108  
Hilman  
Laki-laki  
55  
3 107  
106  
Julia  
Perempuan  
45  
0  
107  
Firman  
Laki-laki  
55  
  
106  
Julia  
Perempuan  
45  
  
108  
Hilman  
Laki-laki  
55
```

11. Kesimpulan

Kesimpulan

Dengan melakukan praktikum ini dapat memahami serta mempraktikkan penggunaan double linked list, pada umumnya double linked list tidak berbeda jauh dengan single linked list. Double linked list selain mencakup data dan next address double linked list juga terdapat previous address yang menyimpan data. Jika di ilustrasikan maka double linked list adalah sebagai berikut:



Melalui praktikum dibuatkan sebuah program yang mengimplementasikan double linked list untuk mendata program haji yang didalamnya dibuatkan struct linked list yang berisikan data (id, nama, gender, usia), next address, dan prev address. Dengan double linked list tersebut dibuatkan beberapa function untuk menambah data di awal, diakhir, dipertengahan(after & before), menghapus data, kemudian mencetak sebuah data.