

Nama : Arya Maulana
Kelas : IF 03-02
NIM : 1203230120

Tugas OTH Double Circular

➤ Source Code dan penjelasan

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  struct Node
6  {
7      int data;
8      struct Node *prev;
9      struct Node *next;
10 };
11
12 typedef struct Node node;
13
14 node *head = NULL;
15 node *tail = NULL;
```

Pada program diatas menggunakan 3 header yang header pertama untuk operasi input output dasar, header kedua untuk operasi pembandingan, dan header ketiga untuk tipe data boolean. Struct node memiliki tiga variabel, variabel data sebagai nilai yang disimpan dalam node dan 'prev' dan 'next' sebagai pointer. Typedef struct digunakan agar lebih mudah untuk memanggil struct node. Node head sebagai posisi awal node dan tail posisi akhir node.

```
16
17 node *createNode()
18 {
19     node *add = NULL;
20     add = (node *)malloc(sizeof(node));
21     return (add);
22 }
```

Fungsi createNode untuk membuat node baru serta mengalokasikan memori menggunakan malloc. Pointer add untuk menyimpan node baru dan mengembalikan fungsinya.

```

23
24 void insert(int data)
25 {
26     node *add = createNode();
27
28     if(add == NULL)
29     {
30         printf("Kosong\n");
31     }
32     else
33     {
34         add->data = data;
35         add->prev = NULL;
36         add->next = NULL;
37
38         if(head == NULL)
39         {
40             head = add;
41             tail = add;
42             head->next = head;
43             head->prev = head;
44         }
45         else
46         {
47             add->prev = tail;
48             add->next = head;
49             tail->next = add;
50             head->prev = add;
51             tail = add;
52         }
53     }
54 }
55

```

Fungsi insert menambahkan data atau nilai pada node baru yang dibuat. Jika node masih belum dibuat maka akan mencetak printf. Jika tidak kosong, maka akan menambahkan data dan node baru diatur sebagai 'tail'.

```

55
56 void view()
57 {
58     node *val = head;
59     int i = 1;
60
61     if(val == NULL)
62     {
63         printf("Kosong\n");
64     }
65     else
66     {
67         printf("\n");
68         while(val != tail)
69         {
70             printf("%d ", val->data);
71             i++;
72             val = val->next;
73         }
74         printf("%d ", val->data);
75     }
76     printf("\n");
77 }
78

```

Fungsi view mencetak semua data pada node dalam linked list. Jika nodenya kosong maka mencetak tulisan pada printf. Sebaliknya, jika ada akan mencetak setiap node yang telah dibuat.

```

78 void sort(node *val, node *afVal)
79 {
80     node *temp = NULL;
81
82     if (val->data > afVal->data)
83     {
84         if (val == head)
85         {
86             head = afVal;
87         }
88         if (afVal == tail)
89         {
90             tail = val;
91         }
92
93         if (val->prev != NULL)
94         {
95             val->prev->next = afVal;
96         }
97         if (afVal->next != NULL)
98         {
99             afVal->next->prev = val;
100         }
101
102         temp = afVal->next;
103         afVal->next = val;
104         afVal->prev = val->prev;
105         val->next = temp;
106         val->prev = afVal;
107     }
108 }
109
110

```

Fungsi sort mengubah posisi dua node yaitu 'val' dan 'afVal' data yang lebih kecil akan diubah diposisi awal. Data mulai dari terkecil hingga terbesar. Data terkecil termasuk head dan data terbesar termasuk tail.

```

111 void address()
112 {
113     node *val = head;
114     int i = 1;
115
116     if (val == NULL)
117     {
118         printf("Kosong\n");
119     }
120     else
121     {
122         printf("\n");
123         while (val != tail)
124         {
125             printf("Address: %p | Data: %d\n ", val, val->data);
126             i++;
127             val = val->next;
128         }
129         printf("Address: %p | Data: %d\n ", val, val->data);
130     }
131     printf("\n");
132 }
133

```

Fungsi address mencetak alamat memori dari setiap data pada node. Val sebagai pointer head dan tail.

```

133
134 int main()
135 {
136     node *add = NULL;
137     int jumlah, data;
138
139     printf("Jumlah data: ");
140     scanf("%d", &jumlah);
141     for(int i = 0; i < jumlah; i++)
142     {
143         printf("Masukkan data ke-%d: ", i + 1);
144         scanf("%d", &data);
145         insert(data);
146     }
147
148     printf("Data sebelum diurutkan:\n");
149     address();
150     printf("Data setelah diurutkan:\n");
151     sort(head, head->next);
152     address();
153
154     return 0;
155 }

```

Fungsi main membuat jumlah data yang akan diinput. Menampilkan data sebelum diascending dan setelah diascending dan memanggil fungsi-fungsi yang telah dibuat.

➤ Ouput(Hasil)

```

PS C:\Users\maula\OneDrive\Documents\algo> cd "c:\Users\maula\OneDrive\Documents\algo" ; if ($?) { gcc othdoublecircular.c -o othdoublecircular } ; if ($?) {
.\othdoublecircular }
Jumlah data: 6
Masukkan data ke-1: 5
Masukkan data ke-2: 5
Masukkan data ke-3: 3
Masukkan data ke-4: 8
Masukkan data ke-5: 1
Masukkan data ke-6: 6
Data sebelum diurutkan:
Address: 000715F0 | Data: 5
Address: 00071598 | Data: 5
Address: 00071580 | Data: 3
Address: 000715C8 | Data: 8
Address: 000723A0 | Data: 1
Address: 000723B8 | Data: 6
Data setelah diurutkan:
Address: 000715F0 | Data: 5
Address: 00071598 | Data: 5
Address: 00071580 | Data: 3
Address: 000715C8 | Data: 8
Address: 000723A0 | Data: 1
Address: 000723B8 | Data: 6
Address: 000723B8 | Data: 6
PS C:\Users\maula\OneDrive\Documents\algo> cd "c:\Users\maula\OneDrive\Documents\algo" ; if ($?) { gcc othdoublecircular.c -o othdoublecircular } ; if ($?) {
.\othdoublecircular }
Jumlah data: 4
Masukkan data ke-1: 3
Masukkan data ke-2: 31
Masukkan data ke-3: 2
Masukkan data ke-4: 123
Data sebelum diurutkan:
Address: 008815F0 | Data: 3
Address: 00881598 | Data: 31
Address: 00881580 | Data: 2
Address: 008815C8 | Data: 123
Data setelah diurutkan:
Address: 008815F0 | Data: 3
Address: 00881598 | Data: 31
Address: 00881580 | Data: 2
Address: 008815C8 | Data: 123
PS C:\Users\maula\OneDrive\Documents\algo>

```