

Nama : Mukhlis Zahrawani Sutrisno

NIM : 1203230065 (IF-03-02)

Tugas : Double Circular ASD

## Tugas Double Circular ASD

Code :

```
D: > C & C++ > C DoublecircularAsd.c > insert(int)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  struct Node
6  {
7      struct Node *prev;
8      int data;
9      struct Node *next;
10 };
11
12 typedef struct Node node;
13
14 node *pHead = NULL;
15 node *pTail = NULL;
16
17 node *alokasiNodeBaru()
18 {
19     node *pNew = NULL;
20     pNew = (node *)malloc(sizeof(node));
21     return (pNew);
22 }
23
```

**Penjelasan :** Program dimulai dengan mendeklarasikan struktur Node. Setiap node ini memiliki tiga bagian, prev yang merupakan pointer ke node sebelumnya, data yang menyimpan nilai data, dan next yang merupakan pointer ke node berikutnya. Kemudian, ada deklarasi tipe data node yang merupakan alias dari struct Node, dan variabel global pHead dan pTail yang merupakan pointer ke node pertama dan terakhir.

Code :

```
D: > C & C++ > C DoublecircularAsd.c > view()
23
24 void insert(int data)
25 {
26     node *pNew = alokasiNodeBaru();
27
28     if (pNew == NULL)
29     {
30         printf("\n[ALOKASI GAGAL]");
31     }
32     else
33     {
34         pNew->data = data;
35         pNew->prev = NULL;
36         pNew->next = NULL;
37
38         if (pHead == NULL)
39         {
40             pHead = pNew;
41             pTail = pNew;
42             pHead->next = pHead;
43             pHead->prev = pHead;
44         }
45         else
46         {
47             pNew->prev = pTail;
48             pNew->next = pHead;
49             pTail->next = pNew;
50             pHead->prev = pNew;
51             pTail = pNew;
52         }
53     }
54 }
55
```

**Penjelasan :** *alokasiNodeBaru* ini berfungsi yang digunakan untuk mengalokasikan memori untuk node baru. Fungsi ini mengembalikan pointer ke node yang dialokasikan

Code :

```
D: > C & C++ > C DoublecircularAsd.c > sortNode(node *, node *)
55
56 void view()
57 {
58     node *pWalker = pHead;
59     int i = 1;
60
61     if (pWalker == NULL)
62     {
63         printf("\n[DATA KOSONG]");
64     }
65     else
66     {
67         printf("\n");
68         while (pWalker != pTail)
69         {
70             printf("%d ", pWalker->data);
71             i++;
72             pWalker = pWalker->next;
73         }
74         printf("%d ", pWalker->data);
75     }
76     printf("\n");
77 }
78
```

**Penjelasan :** `void view` ini berfungsi ini digunakan untuk menampilkan semua data. Program akan menampilkan pesan bahwa data kosong jika linked list kosong. Jika tidak, data akan ditampilkan dari `pHead` hingga `pTail`.

Code :

```
D: > C & C++ > C DoublecircularAsd.c > viewWithAddress()
78
79 void sortNode(node *pWalker, node *pWalkerNext)
80 {
81     node *temp = NULL;
82
83     if (pWalker->data > pWalkerNext->data)
84     {
85         if (pWalker == pHead)
86         {
87             pHead = pWalkerNext;
88         }
89         if (pWalkerNext == pTail)
90         {
91             pTail = pWalker;
92         }
93
94         if (pWalker->prev != NULL)
95         {
96             pWalker->prev->next = pWalkerNext;
97         }
98         if (pWalkerNext->next != NULL)
99         {
100             pWalkerNext->next->prev = pWalker;
101         }
102
103         temp = pWalkerNext->next;
104         pWalkerNext->next = pWalker;
105         pWalkerNext->prev = pWalker->prev;
106         pWalker->next = temp;
107         pWalker->prev = pWalkerNext;
108     }
109 }
110
```

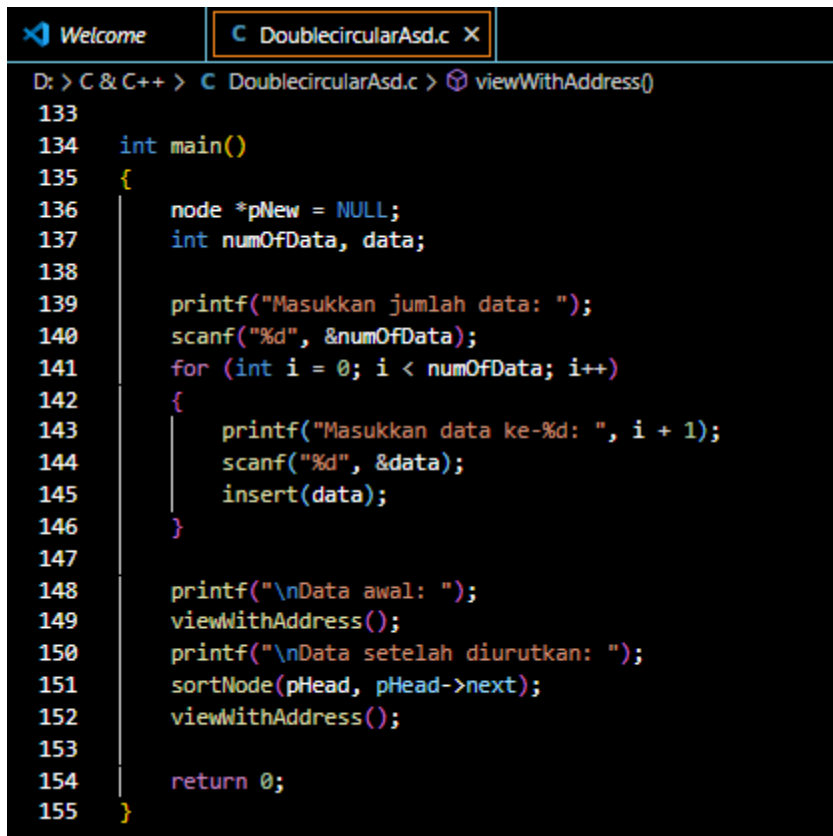
**Penjelasan :** *sortNode* ini berfungsi yang ingin diperbaiki. Fungsi ini seharusnya mengurutkan, tetapi saat ini hanya menukar posisi dua node. Anda perlu mengimplementasikan algoritma pengurutan yang lengkap, seperti algoritma bubble sort, untuk mengurutkan seluruh linked list.

Code :

```
D: > C & C++ > C DoublecircularAsd.c > main()
110
111 void viewWithAddress()
112 {
113     node *pWalker = pHead;
114     int i = 1;
115
116     if (pWalker == NULL)
117     {
118         printf("\n[DATA KOSONG]");
119     }
120     else
121     {
122         printf("\n");
123         while (pWalker != pTail)
124         {
125             printf("Address: %p | Data: %d\n ", pWalker, pWalker->data);
126             i++;
127             pWalker = pWalker->next;
128         }
129         printf("Address: %p | Data: %d\n ", pWalker, pWalker->data);
130     }
131     printf("\n");
132 }
133
```

Penjelasan : *viewWithAddress* ini berfungsi ini mirip dengan *view()*, tetapi juga menampilkan alamat memori dari setiap node.

Code :



```
133
134 int main()
135 {
136     node *pNew = NULL;
137     int numOfData, data;
138
139     printf("Masukkan jumlah data: ");
140     scanf("%d", &numOfData);
141     for (int i = 0; i < numOfData; i++)
142     {
143         printf("Masukkan data ke-%d: ", i + 1);
144         scanf("%d", &data);
145         insert(data);
146     }
147
148     printf("\nData awal: ");
149     viewWithAddress();
150     printf("\nData setelah diurutkan: ");
151     sortNode(pHead, pHead->next);
152     viewWithAddress();
153
154     return 0;
155 }
```

Penjelasan : Fungsi utama dari program. *Int* meminta pengguna untuk memasukkan jumlah data dan data itu sendiri. Kemudian, itu memanggil fungsi *insert* untuk memasukkan data ke dalam linked list. Setelah itu, itu menampilkan data awal dan kemudian mencoba untuk mengurutkan data menggunakan fungsi *sortNode* yang salah.

Hasi Input 01 :

```
PS C:\Users\mukhl> cd "d:\C & C++\" ; if ($?) { gcc DoublecircularAsd.c -o DoublecircularAsd } ; if ($?) { .\DoublecircularAsd }
Masukkan 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 awal:
Address: 00BC2F40 | Data: 5
Address: 00BC2F58 | Data: 5
Address: 00BC2F70 | Data: 3
Address: 00BC0C60 | Data: 8
Address: 00BC0C78 | Data: 1
Address: 00BC0C90 | Data: 6

Data setelah diurutkan:
Address: 00BC2F40 | Data: 5
Address: 00BC2F58 | Data: 5
Address: 00BC2F70 | Data: 3
Address: 00BC0C60 | Data: 8
Address: 00BC0C78 | Data: 1
Address: 00BC0C90 | Data: 6
```

Hasil Input 02 :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\C & C++> cd "d:\C & C++\" ; if ($?) { gcc DoublecircularAsd.c -o DoublecircularAsd } ; if ($?) { .\DoublecircularAsd }
Masukkan jumlah data: 4
Masukkan data ke-1: 3
Masukkan data ke-2: 31
Masukkan data ke-3: 2
Masukkan data ke-4: 123

Data awal:
Address: 00762F40 | Data: 3
Address: 00762F58 | Data: 31
Address: 00762F70 | Data: 2
Address: 00760C60 | Data: 123

Data setelah diurutkan:
Address: 00762F40 | Data: 3
Address: 00762F58 | Data: 31
Address: 00762F70 | Data: 2
Address: 00760C60 | Data: 123
```

Full SS Input :

```
PS C:\Users\mukhl> cd "d:\C & C++\" ; if ($?) { gcc DoublecircularAsd.c -o DoublecircularAsd } ; if ($?) { .\DoublecircularAsd }
Masukkan 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 awal:
Address: 00BC2F40 | Data: 5
Address: 00BC2F58 | Data: 5
Address: 00BC2F70 | Data: 3
Address: 00BC0C60 | Data: 8
Address: 00BC0C78 | Data: 1
Address: 00BC0C90 | Data: 6

Data setelah diurutkan:
Address: 00BC2F40 | Data: 5
Address: 00BC2F58 | Data: 5
Address: 00BC2F70 | Data: 3
Address: 00BC0C60 | Data: 8
Address: 00BC0C78 | Data: 1
Address: 00BC0C90 | Data: 6

PS D:\C & C++> 4
4
PS D:\C & C++> cd "d:\C & C++\" ; if ($?) { gcc DoublecircularAsd.c -o DoublecircularAsd } ; if ($?) { .\DoublecircularAsd }
Masukkan jumlah data: 4
Masukkan data ke-1: 3
Masukkan data ke-2: 31
Masukkan data ke-3: 2
Masukkan data ke-4: 123

Data awal:
Address: 00762F40 | Data: 3
Address: 00762F58 | Data: 31
Address: 00762F70 | Data: 2
Address: 00760C60 | Data: 123
```