Nama : Ahmad Wahyudi

NIM : 1203230116

Kelas : IF 03-02

Double Circular

Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

struct Node
{
    struct Node *prev;
    int data;
    struct Node *next;
};

typedef struct Node node;

node *pHead = NULL;
node *pTail = NULL;

node *alokasiNodeBaru()
{
    node *pNew = NULL;
    pNew = (node *)malloc(sizeof(node));
    return (pNew);
}

void insert(int data)
{
    node *pNew = alokasiNodeBaru();

    if (pNew == NULL)
    {
        printf("\n[ALOKASI GAGAL]");
    }
    else
    {
        pNew->data = data;
        pNew->prev = NULL;
        pNew->next = NULL;
```

```
        if (pHead == NULL)
        {
            pHead = pNew;
            pTail = pNew;
            pHead->next = pHead;
            pHead->prev = pHead;
        }
        else
        {
            pNew->prev = pTail;
            pNew->next = pHead;
            pTail->next = pNew;
            pHead->prev = pNew;
            pTail = pNew;
        }
    }
}

void view()
{
    node *pWalker = pHead;
    int i = 1;

    if (pWalker == NULL)
    {
        printf("\n[DATA KOSONG]");
    }
    else
    {
        printf("\n");
        while (pWalker != pTail)
        {
            printf("%d ", pWalker->data);
            i++;
            pWalker = pWalker->next;
        }
        printf("%d ", pWalker->data);
    }
    printf("\n");
}

void sortNode(node *pWalker, node *pWalkerNext)
{
    node *temp = NULL;
```

```
    if (pWalker->data > pWalkerNext->data)
    {
        if (pWalker == pHead)
        {
            pHead = pWalkerNext;
        }
        if (pWalkerNext == pTail)
        {
            pTail = pWalker;
        }

        if (pWalker->prev != NULL)
        {
            pWalker->prev->next = pWalkerNext;
        }
        if (pWalkerNext->next != NULL)
        {
            pWalkerNext->next->prev = pWalker;
        }

        temp = pWalkerNext->next;
        pWalkerNext->next = pWalker;
        pWalkerNext->prev = pWalker->prev;
        pWalker->next = temp;
        pWalker->prev = pWalkerNext;
    }
}

void viewWithAddress()
{
    node *pWalker = pHead;
    int i = 1;

    if (pWalker == NULL)
    {
        printf("\n[DATA KOSONG]");
    }
    else
    {
        printf("\n");
        while (pWalker != pTail)
        {
            printf("Address: %p | Data: %d\n ", pWalker, pWalker->data);
            i++;
```

```
            pWalker = pWalker->next;
        }
        printf("Address: %p | Data: %d\n ", pWalker, pWalker->data);
    }
    printf("\n");
}

int main()
{
    node *pNew = NULL;
    int numOfData, data;

    printf("Masukkan jumlah data: ");
    scanf("%d", &numOfData);
    for (int i = 0; i < numOfData; i++)
    {
        printf("Masukkan data ke-%d: ", i + 1);
        scanf("%d", &data);
        insert(data);
    }

    printf("\nData awal: ");
    viewWithAddress();
    printf("\nData setelah diurutkan: ");
    sortNode(pHead, pHead->next);
    viewWithAddress();

    return 0;
}
```

Penjelasan

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

struct Node
{
    struct Node *prev;
    int data;
    struct Node *next;
};

typedef struct Node node;

node *pHead = NULL;
node *pTail = NULL;

node *alokasiNodeBaru()
{
    node *pNew = NULL;
    pNew = (node *)malloc(sizeof(node));
    return (pNew);
}

```

- stdio.h berfungsi sebagai input dan output.
  stdlib.h berfungsi mengatur memori.
  stdbool berfungsi untuk mendefinisikan tipe data bool yaitu true dan false.
- struct node mendefinisikan node dalam linked list dengan komponen
  prev, data, next.
- typedef berfungsi mendefinisikan node sebagai alias untuk struct node, sehingga kita bisa menggunakan node sebagai tipe data yang lebih singkat.
- alokasinodebaru berfungsi mengalokasikan memori untuk node baru

```c
void insert(int data)
{
    node *pNew = alokasiNodeBaru();

    if (pNew == NULL)
    {
        printf("\n[ALOKASI GAGAL]");
    }
    else
    {
        pNew->data = data;
        pNew->prev = NULL;
        pNew->next = NULL;

        if (pHead == NULL)
        {
            pHead = pNew;
            pTail = pNew;
            pHead->next = pHead;
            pHead->prev = pHead;
        }
        else
        {
            pNew->prev = pTail;
            pNew->next = pHead;
            pTail->next = pNew;
            pHead->prev = pNew;
            pTail = pNew;
        }
    }
}

```

- void insert berfungsi untuk menambahkan node baru ke dalam double circular linked list, data akan ditempatkan di node pertama jika linked list kosong dan ditaruh di akhir jika node sudah berisi.

```
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
```

- void view berfungsi memastikan semua node ditampilkan dengan benar, dari node pertama phead sampai terakhir ptail.

```
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
```

- void sortnode berfungsi menukar node yang berdekatan untuk mengurutkan nilai data.

```
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
```

- viewwithaddress berfugsi untuk memahami struktur dan urutan node dalam linked list dengan melihat data dan alamat setiap node yang disimpan.

```
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  }
```

- int main berfungsi meminta user memasukan data dalam linked list, program menampilkan data awal dan data setelah diurutkan, serta menampilkan alamat memori dari setiap node.

Output

```
PS C:\Users\Ahmad Wahyudi> cd "d:\program\Alpro S2\" ; if ($?) { gcc doublesingular.c -o d
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: 00C81390 | Data: 5
 Address: 00C813A8 | Data: 5
 Address: 00C813C0 | Data: 3
 Address: 00C813D8 | Data: 8
 Address: 00C813F0 | Data: 1
 Address: 00C81408 | Data: 6


Data setelah diurutkan:
Address: 00C81390 | Data: 5
 Address: 00C813A8 | Data: 5
 Address: 00C813C0 | Data: 3
 Address: 00C813D8 | Data: 8
 Address: 00C813F0 | Data: 1
 Address: 00C81408 | Data: 6
```

```
PS C:\Users\Ahmad Wahyudi> cd "d:\program\Alpro S2\" ; if ($?)
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: 00C51390 | Data: 3
 Address: 00C513A8 | Data: 31
 Address: 00C513C0 | Data: 2
 Address: 00C513D8 | Data: 123


Data setelah diurutkan:
Address: 00C51390 | Data: 3
 Address: 00C513A8 | Data: 31
 Address: 00C513C0 | Data: 2
 Address: 00C513D8 | Data: 123

PS D:\program\Alpro S2>
```