

Nama : Insan Nafi' Alkalas
Kelas: IF 03-03
NIM: 1203230022

Double Linked list Circular

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

```
typedef struct NODE *node;
```

```
struct NODE
{
    int value_node;
    node next;
    node prev;
};
```

```
node head = NULL;
```

```
node create_node(int value)
{
    node local = (node)malloc(sizeof(node));
    local->value_node = value;
    local->next = local->prev = head;
    return local;
}
```

```
void insert_first(int value)
{
    node node_local = create_node(value);
```

```
(head == NULL) ? (head = node_local, head->next = head->prev = node_local) : (node_local->next = head, node_local->prev = head->prev, head->prev->next = node_local, head = node_local);
}
```

```
void insert_last(int value)
{
    node node_local = create_node(value);
```

```
(head == NULL) ? (head = node_local, head->next = head->prev = node_local) : (node_local->next = head, node_local->prev = head->prev, head->prev->next = node_local, head->prev = node_local);
}
```

```
void swap (node *head, node a, node b)
{
    if (a == b) return; // No need to swap identical nodes
```

```
node a_prev = a->prev;
node a_next = a->next;
node b_prev = b->prev;
node b_next = b->next;
```

```
if (a->next == b)
{
    // a and b are adjacent with a before b
    a->next = b_next;
    a->prev = b;
    b->next = a;
    b->prev = a_prev;
    a_prev->next = b;
    b_next->prev = a;
} else if (b->next == a)
{
    // a and b are adjacent with b before a
    b->next = a_next;
    b->prev = a;
    a->next = b;
    a->prev = b_prev;
    b_prev->next = a;
    a_next->prev = b;
} else
{
    // a and b are not adjacent
    a->next = b_next;
    a->prev = b_prev;
    b->next = a_next;
    b->prev = a_prev;
    a_prev->next = b;
    a_next->prev = b;
    b_prev->next = a;
    b_next->prev = a;
}
```

```
if (*head == a) *head = b;
else if (*head == b) *head = a;
}
```

```
void sort_list()
{
    if(head == NULL) return;
```

```
node current;
int temp;
```

```
do
{
```

```

temp = 0;
current = head;

do
{
if (current->value_node > current->next->value_node)
{
// tukar node menggunakan fungsi
swap(&head, current, current->next);
// ubah nilai repeat jadi 1 agar bisa melakukan looping dari awal lagi
temp = 1;
} else
{
// jika node tidak memenuhi kriteria, maka lanjut ke node selanjutnya
current = current->next;
}
} while(current->next != head);

} while (temp);
}

```

```

void show_list()
{
node local = head;

if (head == NULL) return;

do
{
printf("Address: %p\nValue: %d\n", local, local->value_node);
local = local->next;

} while (local != head);
printf("\n");
}

```

```

int main()
{
int choice, value_, length, buffer;

do
{
printf("1. Tambahkan di Awal List\n");
printf("2. Tambahkan di Akhir List\n");
printf("3. Tampilkan Nilai List\n");
printf("4. Add with Limitid number\n");
printf("0. Keluar\n");
printf("Pilihan anda: ");

```

```
scanf("%d", &choice);
```

```
switch (choice)
```

```
{
```

```
case 1:
```

```
printf("Masukkan nilai: ");
```

```
scanf("%d", &value_);
```

```
insert_first(value_);
```

```
printf("\n");
```

```
break;
```

```
case 2:
```

```
printf("Masukkan nilai: ");
```

```
scanf("%d", &value_);
```

```
insert_last(value_);
```

```
printf("\n");
```

```
break;
```

```
case 3:
```

```
printf("\nList saat ini:\n");
```

```
show_list();
```

```
printf("\n");
```

```
sort_list();
```

```
show_list();
```

```
printf("\n");
```

```
break;
```

```
case 4:
```

```
printf("Masukan Panjang Linked list: ");
```

```
scanf("%d", &length);
```

```
printf("1. Insert first | 2. Insert Last: ");
```

```
scanf("%d", &value_);
```

```
if (value_ == 1)
```

```
{
```

```
for (int i = 0; i < length; i++)
```

```
{
```

```
scanf("%d", &buffer);
```

```
// tambah node baru menggunakan insertLast
```

```
insert_first(buffer);
```

```
// jika list kosong, maka ubah list first juga
```

```
if (head == NULL) head;
```

```
}
```

```
printf("\nList saat ini:\n");
```

```
show_list();
```

```
printf("\n");
```

```
sort_list();
```

```
show_list();
```

```

printf("\n");

} else
{
for (int i = 0; i < length; i++) {
scanf("%d", &buffer);
// tambah node baru menggunakan insertLast
insert_last(buffer);
// jika list kosong, maka ubah list first juga
if (head == NULL) head;
}

printf("\nList saat ini:\n");
show_list();
printf("\n");
sort_list();
show_list();
printf("\n");
}
return 0;
case 0:
return 0;
default:
break;
}
} while (choice != 0);
return 0;
}

```

Penejelasan:

baris 4: digunakan untuk membuat struktur node / node

baris 6 – 11: digunakan untuk membuat isi struktur dalam node. Yang berisi 3 elemen, yaitu data, next, dan prev.

baris 13: digunakan untuk membaeri nilai/varibel head = NULL

baris 15 – 21: baris ini digunakan untuk membuat node tersebut. Yang dimana node baru akan mengalokasikan memori dengan besarnya memori node. Kemudian node baru akan dikasih elemnt yaitu, data, prev, dan next

baris 23 – 28: baris ini akan memasukan data/nilai mulai dari awal. Dengan cara membuat node baru (memanggil function create_node). Kemudian melakukan pengecekan yang dimana data tersebut NULL atau tidak, jika NULL maka data tersebut akan diisi yang data yang baru dimasukkan, jika tidak maka data akan melanjutkan pengisian dari awal (samping kiri)

baris 30 – 36: baris ini akan memasukan data/nilai mulai dari akhir. Dengan cara membuat node baru (memanggil function create_node). Kemudian melakukan pengecekan yang dimana data tersebut NULL atau tidak, jika NULL maka data tersebut akan diisi yang data yang baru dimasukkan, jika tidak maka data akan melanjutkan pengisian dari akhir (samping kanan)

baris 38 – 80: baris ini digunakan untuk mengganti posisi/swap node beserta alamat dari node tersebut. Yang dimana pada baris ini banyak variabel bantuan yaitu a_prev, a_next, b_prev, dan b_next. Tapi

sebelum melakukan perubahan posisi, data akan melakukan pengecekan yang dimana apakah data tersebut sudah terurut atau belum, kalau sudah data tidak akan melakukan pengurutan jika belum akan melakukan pengecekan lagi, yaitu Apakah data tersebut bersampingan/apakah data tersebut jaraknya berjauhan. Jika data tersebut berdekatan maka akan mengeksekusi code pada baris 47 – 64 jika berjauhan maka akan mengeksekusi code pada baris 65 – 76. Dan setelah melihat jarak jauhnya posisi data akan melakukan perubahan posisi.

Baris 82 – 110: baris/function ini digunakan untuk mengurutkan data yang mulai terkecil hingga terbesar. Pada baris ini data/value yang akan dicari kemudian memanggil function `swap_node/swap` untuk merubah posisinya(jika ada data yang lebih kecil dari data sebelum/sesudah list). Pada sorting ini digunakan bubble sort

Baris 112 – 126: baris/function ini digunakan untuk menampilkan output tersebut yang dimana akan menampilkan nilai data tersebut dan juga alamat memori tersebut sebelum/sesudah disorting.

Baris 128 – 216: baris/function ini digunakan untuk menjalankan perintah program yang dimana isinya beragam. Pada function ini terdapat perintah yang banyak.

Baris 130: baris ini digunakan untuk membuat variabel lokal

baris 132 – 214: baris ini digunakan untuk perulangan yang dimana dalam perulangan tersebut mempunyai code lagi dan jangan sampai menginputkan 0 karena pada baris 214 ada kondisi yang dimana user menginputkan 0 maka program berhenti.

Baris 134 – 139: baris ini hanya sekedar output menu untuk memilih perintah

baris 142 – 213: baris ini digunakan untuk pengecekan pemilihan.

Baris 144 – 150: baris ini digunakan untuk penginputan dengan cara memasukkan data tersebut satu per-satu dan data yang masuk akan dimasukkan ke dalam function `insert_first`

baris 151 – 157: baris ini digunakan untuk penginputan dengan cara memasukkan data tersebut satu per-satu dan data yang masuk akan dimasukkan ke dalam function `insert_last`

baris 158 – 165: baris ini digunakan untuk mengeluarkan inputan yang sudah diinputkan dengan cara memanggil function `show_list`.

Baris 166 – 207: baris ini digunakan untuk melakukan perulangan yang dimana kegunaanya sama dengan baris sebelumnya, cuma yang berbeda di baris ini user perlu untuk melakukan penginputan maksimal penginputannya. Dan di baris 173 akan melakukan pengecekan, yang dimana baris ini hanya lebih ingin menggunakan function apa untuk memasukkan data tersebut. Apakah ingin menggunakan `insert_last/insert_first`. Dan jika sudah menginputkan maka output tersebut akan keluar

output:

```
List saat ini:  
Address: 0x5b88a3bleac0  
Value: 3  
Address: 0x5b88a3bleae0  
Value: 2  
Address: 0x5b88a3bleb00  
Value: 1
```

```
Address: 0x5b88a3bleb00  
Value: 1  
Address: 0x5b88a3bleae0  
Value: 2  
Address: 0x5b88a3bleac0  
Value: 3
```

```
List saat ini:  
Address: 0x6311c6a28ac0  
Value: 4  
Address: 0x6311c6a28ae0  
Value: 3  
Address: 0x6311c6a28b00  
Value: 2  
Address: 0x6311c6a28b20  
Value: 1  
Address: 0x6311c6a28b40  
Value: 5
```

```
Address: 0x6311c6a28b20  
Value: 1  
Address: 0x6311c6a28b00  
Value: 2  
Address: 0x6311c6a28ae0  
Value: 3  
Address: 0x6311c6a28ac0  
Value: 4  
Address: 0x6311c6a28b40  
Value: 5
```