

## Tugas 2 Circular Singly Linked List

Nama : Romadhoni Kusuma Nanda Prihadi

Kelas / NIM : PTI B 2023 / 23050974063

1. Buatlah source code untuk mencetak Elemen dari Circular Singly Linked List dalam Urutan Terbalik !

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;

    Node(int data) : data(data), next(nullptr) {}
};

class CircularLinkedList {
private:
    Node* head;

    void printTerbalik(Node* node, Node* start) const {

        if (node->next == start) {
            cout << node->data << " ";
            return;
        }

        printTerbalik(node->next, start);
        cout << node->data << " ";
    }

public:

    CircularLinkedList() : head(nullptr) {}

    ~CircularLinkedList() {
        if (head) {
            Node* current = head;
            Node* nextNode;
            do {
                nextNode = current->next;
                delete current;
                current = nextNode;
            } while (current != head);
        }
    }
}
```

```

void insertDepan(int data) {
    Node* newNode = new Node(data);
    if (!head) {\
        head = newNode;
        newNode->next = head;
    } else {
        newNode->next = head;
        Node* temp = head;
        while (temp->next != head) {
            temp = temp->next;
        }
        temp->next = newNode;
        head = newNode;
    }
}

void Output() const {
    if (!head) {
        cout << "List tidak ada" << endl;
        return;
    }
    Node* temp = head;
    do {
        cout << temp->data << " ";
        temp = temp->next;
    } while (temp != head);
    cout << endl;
}

void outputTerbalik() const {
    if (!head) {
        cout << "List tidak ada" << endl;
        return;
    }
    printTerbalik(head, head);
    cout << endl;
}

};

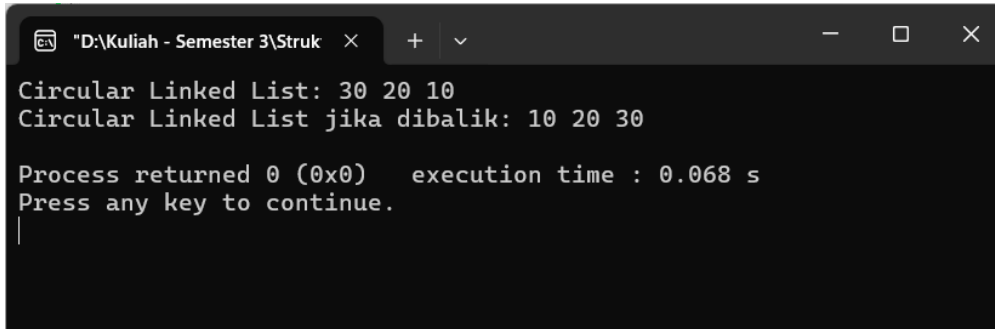
int main() {
    CircularLinkedList cll;
    cll.insertDepan(10);
    cll.insertDepan(20);
    cll.insertDepan(30);

    cout << "Circular Linked List: ";
    cll.Output();
}

```

```
cout << "Circular Linked List jika dibalik: ";  
ccl.outputTerbalik();  
  
return 0;  
}
```

### Output:



```
"D:\Kuliah - Semester 3\Struk" × + ▾  
Circular Linked List: 30 20 10  
Circular Linked List jika dibalik: 10 20 30  
  
Process returned 0 (0x0) execution time : 0.068 s  
Press any key to continue.  
|
```

### Keterangan:

- **Fungsi printTerbalik:**  
Fungsi ini mencetak elemen dari akhir ke awal. Cara kerjanya adalah fungsi berjalan melalui semua node hingga mencapai node terakhir, lalu mencetak nilainya mulai dari node terakhir hingga kembali ke node pertama.
- **Fungsi insertDepan:**  
Fungsi ini untuk menginput elemen di depan.
- **Fungsi Output:**  
Menampilkan elemen list.
- **Fungsi outputTerbalik:**  
Menampilkan elemen list secara terbalik.

2. Buatlah source code untuk menghitung Panjang Circular Singly Linked List !

```
#include <iostream>  
using namespace std;  
  
struct Node {  
    int data;  
    Node* next;  
  
    Node(int data) : data(data), next(nullptr) {}  
};
```

```

class CircularLinkedList {
private:
    Node* head;

public:
    CircularLinkedList() : head(nullptr) {}

    ~CircularLinkedList() {
        if (head) {
            Node* current = head;
            Node* nextNode;
            do {
                nextNode = current->next;
                delete current;
                current = nextNode;
            } while (current != head);
        }
    }

    void insertDepan(int data) {
        Node* newNode = new Node(data);
        if (!head) {
            head = newNode;
            newNode->next = head;
        } else {
            newNode->next = head;
            Node* temp = head;
            while (temp->next != head) {
                temp = temp->next;
            }
            temp->next = newNode;
            head = newNode;
        }
    }

    void Output() const {
        if (!head) {
            cout << "List tidak ada" << endl;
            return;
        }
        Node* temp = head;
        do {
            cout << temp->data << " ";
            temp = temp->next;
        } while (temp != head);
        cout << endl;
    }
}

```

```

int panjang() const {
    if (!head) {
        return 0;
    }
    int count = 0;
    Node* temp = head;
    do {
        count++;
        temp = temp->next;
    } while (temp != head);
    return count;
}

};

int main() {
    CircularLinkedList cll;
    cll.insertDepan(10);
    cll.insertDepan(20);
    cll.insertDepan(30);
    cll.insertDepan(40);
    cll.insertDepan(100);

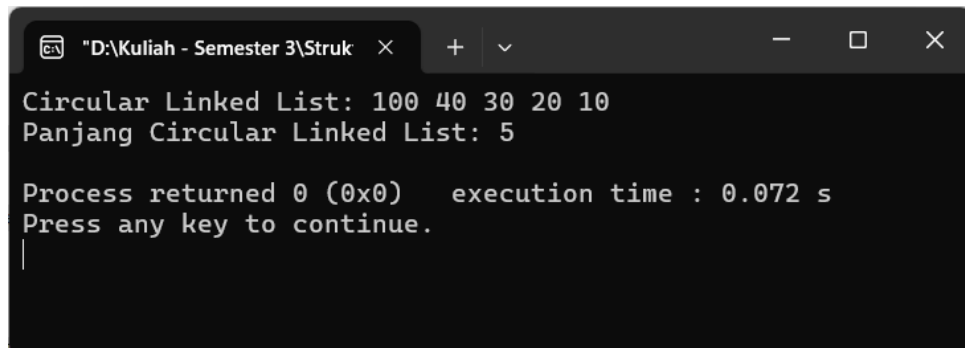
    cout << "Circular Linked List: ";
    cll.Output();

    cout << "Panjang Circular Linked List: " << cll.panjang() << endl;

    return 0;
}

```

### Output:



```

D:\Kuliah - Semester 3\Struk
Circular Linked List: 100 40 30 20 10
Panjang Circular Linked List: 5

Process returned 0 (0x0)   execution time : 0.072 s
Press any key to continue.

```

**Keterangan:**

- **Fungsi insertDepan:**  
Fungsi ini untuk menginput elemen di depan.
- **Fungsi Output:**  
Menampilkan elemen list.
- **Fungsi panjang:**  
Fungsi ini berjalan melalui semua node dan menghitung berapa banyak node yang ada dalam list tersebut.

3. Buatlah source code untuk mencari Elemen di Circular Singly Linked List serta Tambahkan metode untuk mencari elemen dalam circular singly linked list dan mengembalikan apakah elemen tersebut ada atau tidak !

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;

    Node(int data) : data(data), next(nullptr) {}
};

class CircularLinkedList {
private:
    Node* head;

public:
    CircularLinkedList() : head(nullptr) {}

    ~CircularLinkedList() {
        if (head) {
            Node* current = head;
            Node* nextNode;
            do {
                nextNode = current->next;
                delete current;
                current = nextNode;
            } while (current != head);
        }
    }

    void insertDepan(int data) {
        Node* newNode = new Node(data);
        if (!head) {
            head = newNode;
        }
    }
};
```

```

        newNode->next = head;
    } else {
        newNode->next = head;
        Node* temp = head;
        while (temp->next != head) {
            temp = temp->next;
        }
        temp->next = newNode;
        head = newNode;
    }
}

void Output() const {
    if (!head) {
        cout << "List tidak ada" << endl;
        return;
    }
    Node* temp = head;
    do {
        cout << temp->data << " ";
        temp = temp->next;
    } while (temp != head);
    cout << endl;
}

bool search(int hasil) const {
    if (!head) {
        return false;
    }
    Node* temp = head;
    do {
        if (temp->data == hasil) {
            return true;
        }
        temp = temp->next;
    } while (temp != head);
    return false;
}

};

int main() {
    CircularLinkedList cll;
    cll.insertDepan(10);
    cll.insertDepan(20);
    cll.insertDepan(30);
    cll.insertDepan(40);
    cll.insertDepan(100);
}

```

```

cout << "Circular Linked List: ";
ccl.Output();

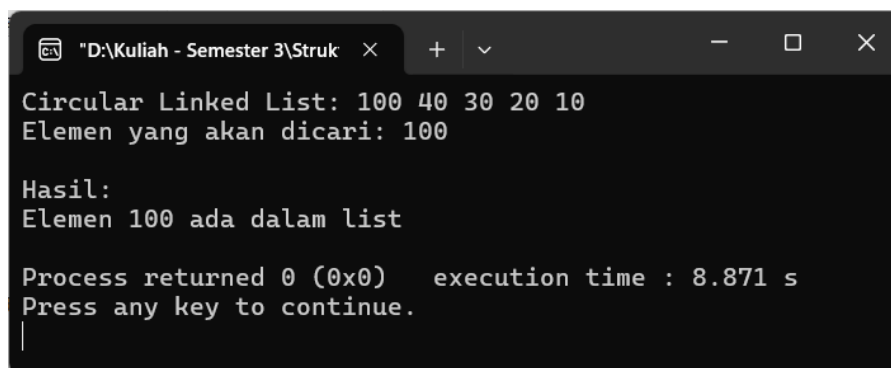
int hasil;
cout << "Elemen yang akan dicari: ";

cin >> hasil;
cout << "\nHasil: \n";
if (ccl.search(hasil)) {
    cout << "Elemen " << hasil << " ada dalam list" << endl;
} else {
    cout << "Elemen " << hasil << " tidak ada dalam list" << endl;
}

return 0;
}

```

#### Output:



```

Circular Linked List: 100 40 30 20 10
Elemen yang akan dicari: 100

Hasil:
Elemen 100 ada dalam list

Process returned 0 (0x0)   execution time : 8.871 s
Press any key to continue.
|

```

#### Keterangan:

- **Fungsi insertDepan:**  
Fungsi ini untuk menginput elemen di depan.
- **Fungsi Output:**  
Menampilkan elemen list.
- **Fungsi search:**  
Fungsi ini menjelajahi Circular Linked List dan membandingkan nilai setiap node dengan nilai (hasil) yang dicari. Jika menemukan elemen yang cocok, fungsi akan mengembalikan nilai true. Jika tidak menemukan elemen tersebut, fungsi akan mengembalikan nilai false.