Tugas : Minggu Ke-3

Nama : Fitri Romadhona

NIM : 23050974179

Kelas : PTI 2023E

Dosen Pengampu : Riza Akhsani Setyo Prayoga, S.Kom., M.MT.

Mata Kuliah : Struktur Data

1. **Source Code**

```
#include <iostream>

using namespace std;

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

// Fungsi untuk membuat node baru
Node* newNode(int data) {
   Node* node = new Node;
   node->data = data;
   node->next = NULL;
   return node;
}

// Fungsi untuk mencetak elemen-elemen linked list
void printList(Node* head) {
   if (head == NULL) {
      cout << "Linked list kosong" << endl;
      return;
   }

   Node* temp = head;
   do {
      cout << temp->data << " ";
      temp = temp->next;
   } while (temp != head);

   cout << endl;
}

// Fungsi untuk membalikkan linked list circular
Node* reverseCircularLinkedList(Node* head) {
   if (head == NULL || head->next == head) {
      return head;
   }
```

```
    Node* current = head;
    Node* prev = NULL;
    Node* next = NULL;

    do {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    } while (current != head);

    head->next = prev;
    head = prev;

    return head;
}

// Fungsi untuk menghitung panjang linked list circular
int countNodes(Node* head) {
    if (head == NULL) {
        return 0;
    }

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

    return count;
}

// Fungsi untuk mencari elemen di linked list circular
bool searchNode(Node* head, int data) {
    if (head == NULL) {
        return false;
    }

    Node* temp = head;
    do {
        if (temp->data == data) {
            return true;
        }
        temp = temp->next;
    } while (temp != head);

    return false;
}
```

```
int main() {
    // Membuat linked list circular
    Node* head = newNode(12);
    head->next = newNode(56);
    head->next->next = newNode(2);
    head->next->next->next = head;

    // Menampilkan linked list asli
    cout << "Linked list asli: ";
    printList(head);

    // Membalikkan linked list circular
    head = reverseCircularLinkedList(head);

    // Menampilkan linked list terbalik
    cout << "Linked list terbalik: ";
    printList(head);

    // Menghitung panjang linked list circular
    int length = countNodes(head);
    cout << "Panjang linked list circular: " << length << endl;

    // Mencari elemen di linked list circular
    int data = 56;
    bool found = searchNode(head, data);
    if (found) {
        cout << data << " ditemukan di linked list circular" << endl;
    } else {
        cout << data << " tidak ditemukan di linked list circular" << endl;
    }

    return 0;
}
```

2. Hasil yang Ditampilkan