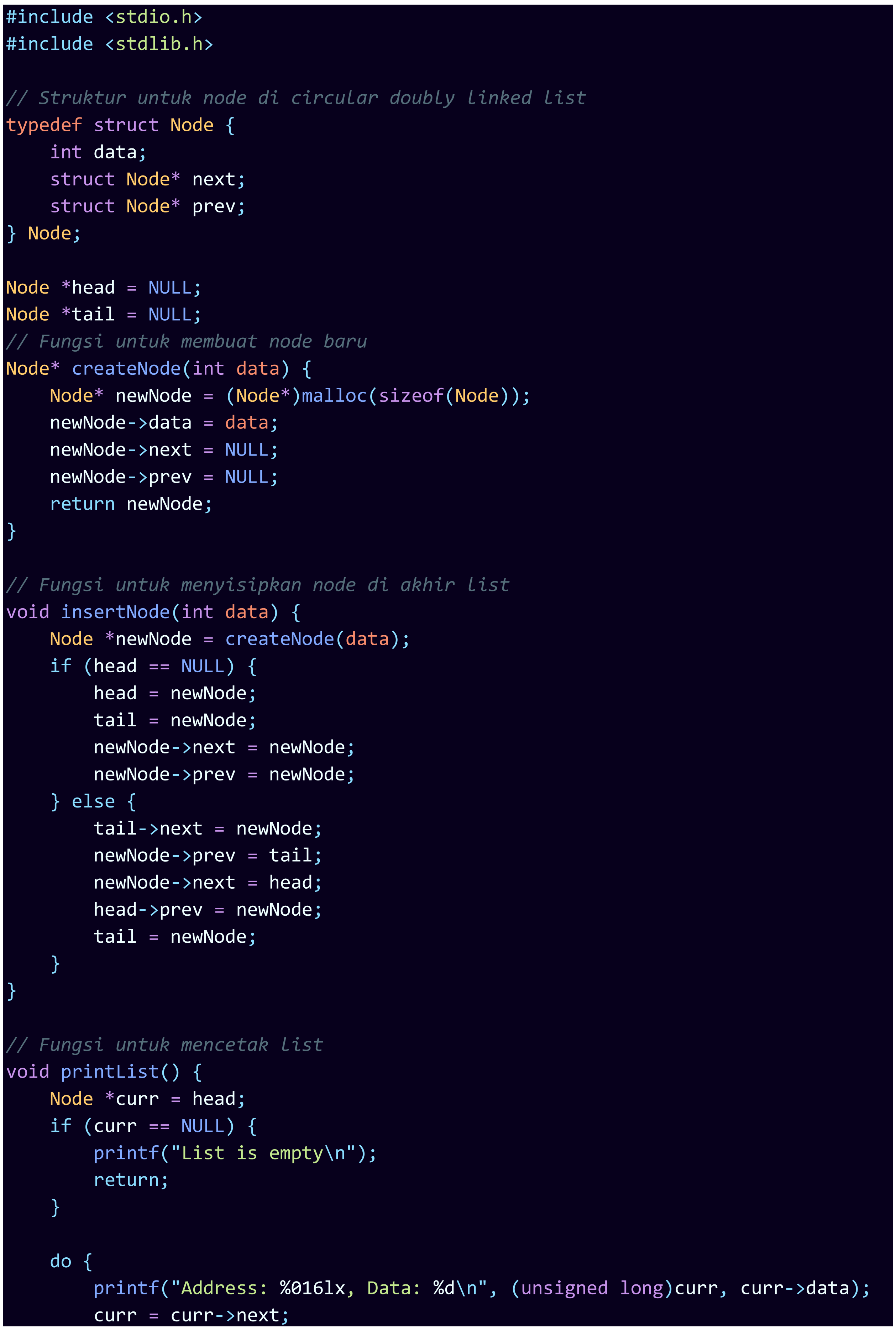
# TUGAS OTH

Nama : Naufal Fahmi Ahnaf

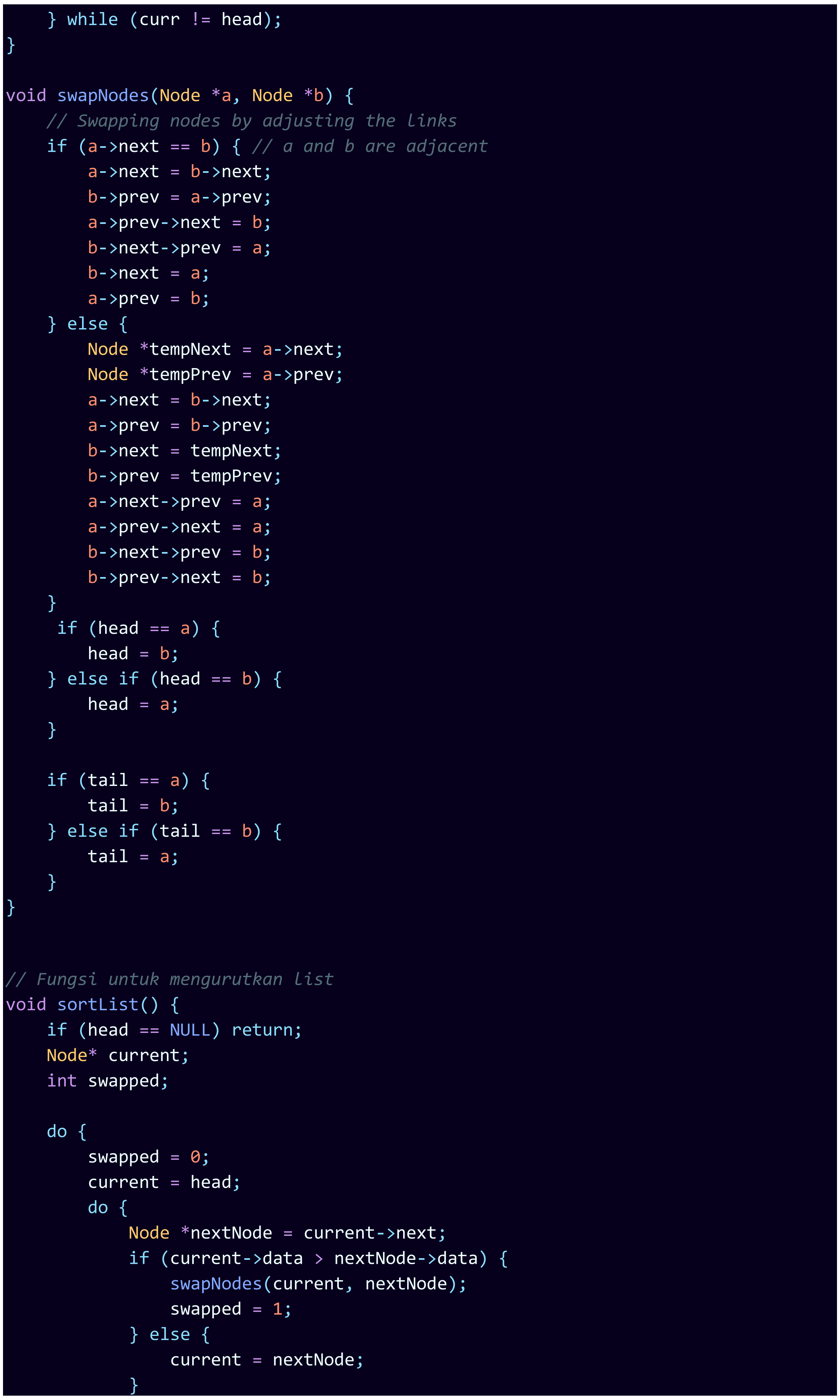
Kelas : IF – 03 – 01

NIM : 1203230034

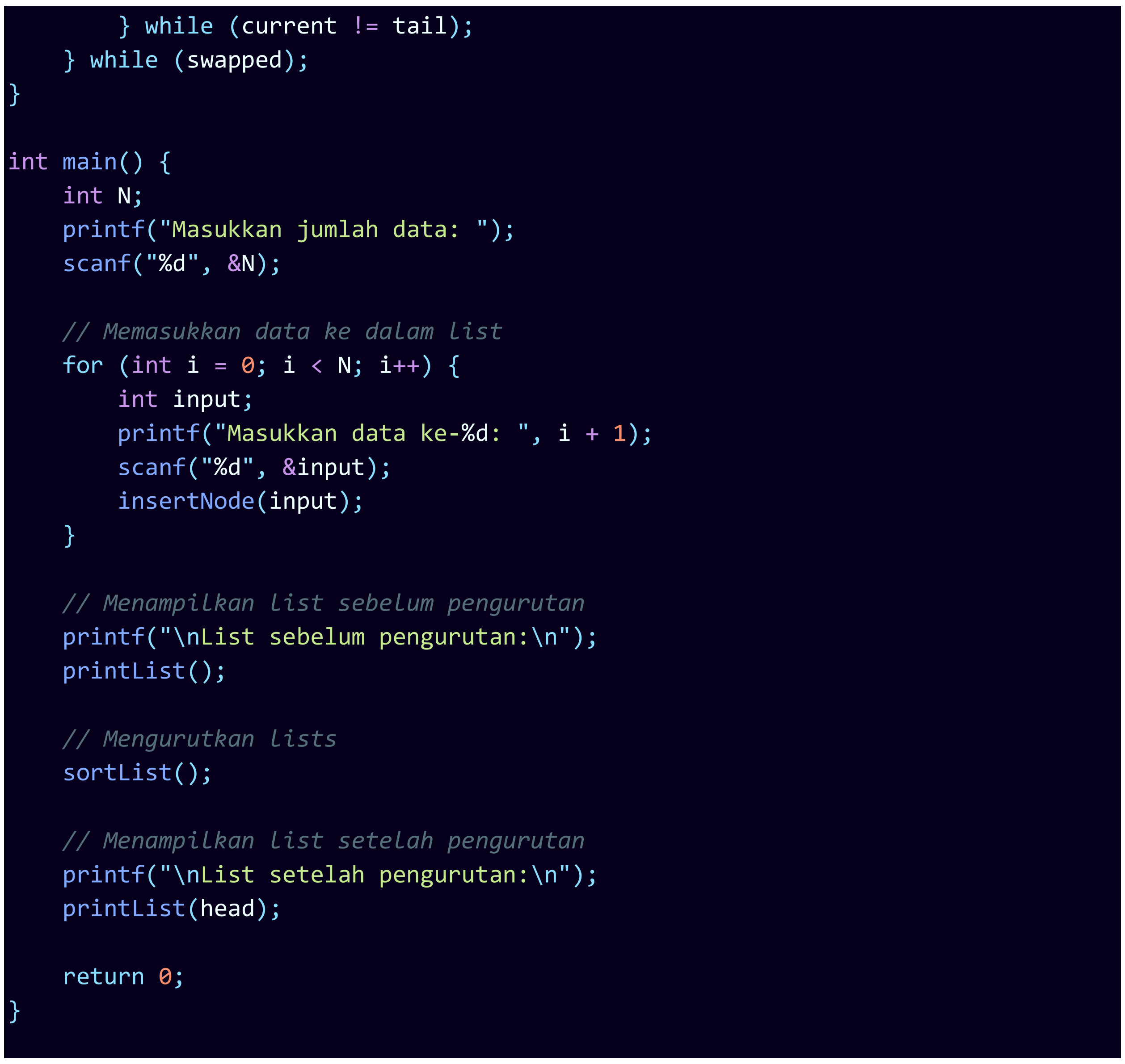
1.Source code & penjelasan setiap linenya



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

// Struktur untuk node di circular doubly linked list
typedef struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
} Node;

Node *head = NULL;
Node *tail = NULL;
// Fungsi untuk membuat node baru
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

// Fungsi untuk menyisipkan node di akhir list
void insertNode(int data) {
    Node *newNode = createNode(data);
    if (head == NULL) {
        head = newNode;
        tail = newNode;
        newNode->next = newNode;
        newNode->prev = newNode;
    } else {
        tail->next = newNode;
        newNode->prev = tail;
        newNode->next = head;
        head->prev = newNode;
        tail = newNode;
    }
}

// Fungsi untuk mencetak list
void printList() {
    Node *curr = head;
    if (curr == NULL) {
        printf("List is empty\n");
        return;
    }

    do {
        printf("Address: %016lx, Data: %d\n", (unsigned long)curr, curr->data);
        curr = curr->next;
```

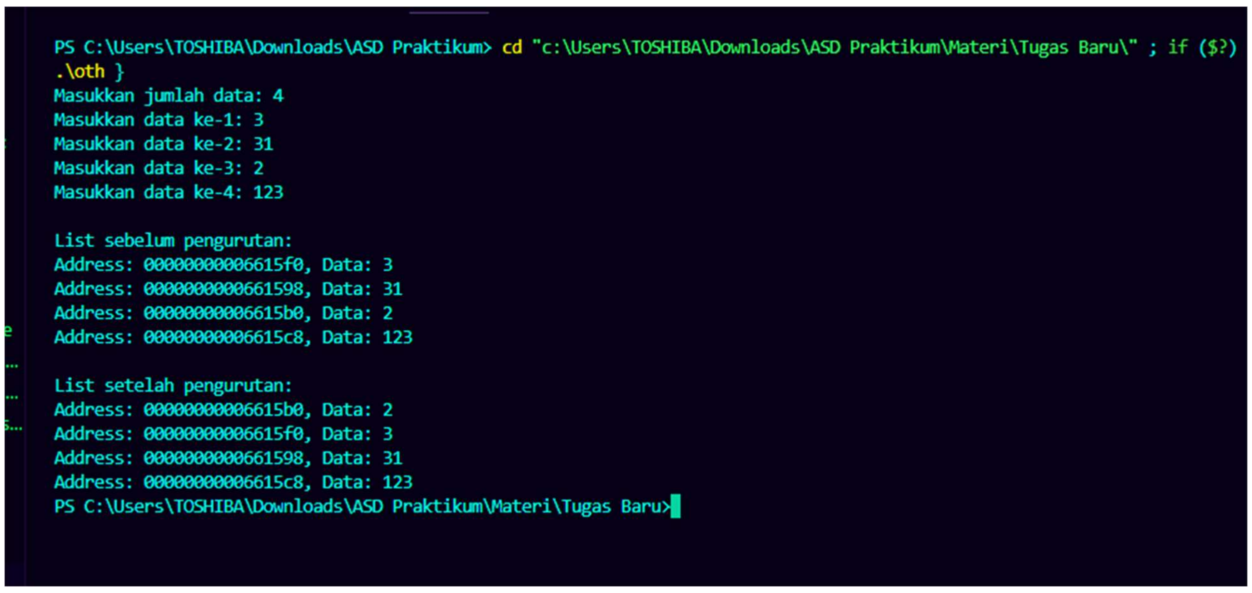


```
    } while (curr != head);
}

void swapNodes(Node *a, Node *b) {
    // Swapping nodes by adjusting the links
    if (a->next == b) { // a and b are adjacent
        a->next = b->next;
        b->prev = a->prev;
        a->prev->next = b;
        b->next->prev = a;
        b->next = a;
        a->prev = b;
    } else {
        Node *tempNext = a->next;
        Node *tempPrev = a->prev;
        a->next = b->next;
        a->prev = b->prev;
        b->next = tempNext;
        b->prev = tempPrev;
        a->next->prev = a;
        a->prev->next = a;
        b->next->prev = b;
        b->prev->next = b;
    }
     if (head == a) {
        head = b;
    } else if (head == b) {
        head = a;
    }

    if (tail == a) {
        tail = b;
    } else if (tail == b) {
        tail = a;
    }
}


// Fungsi untuk mengurutkan list
void sortList() {
    if (head == NULL) return;
    Node* current;
    int swapped;

    do {
        swapped = 0;
        current = head;
        do {
            Node *nextNode = current->next;
            if (current->data > nextNode->data) {
                swapNodes(current, nextNode);
                swapped = 1;
            } else {
                current = nextNode;
            }
```



```
        } while (current != tail);
    } while (swapped);
}

int main() {
    int N;
    printf("Masukkan jumlah data: ");
    scanf("%d", &N);

    // Memasukkan data ke dalam list
    for (int i = 0; i < N; i++) {
        int input;
        printf("Masukkan data ke-%d: ", i + 1);
        scanf("%d", &input);
        insertNode(input);
    }

    // Menampilkan list sebelum pengurutan
    printf("\nList sebelum pengurutan:\n");
    printList();

    // Mengurutkan lists
    sortList();

    // Menampilkan list setelah pengurutan
    printf("\nList setelah pengurutan:\n");
    printList(head);

    return 0;
}
```

2. Output



```
PS C:\Users\TOSHIBA\Downloads\ASD Praktikum> cd "c:\Users\TOSHIBA\Downloads\ASD Praktikum\Materi\Tugas Baru\" ; if ($?)
.\oth }
Masukkan jumlah data: 4
Masukkan data ke-1: 3
Masukkan data ke-2: 31
Masukkan data ke-3: 2
Masukkan data ke-4: 123

List sebelum pengurutan:
Address: 00000000006615f0, Data: 3
Address: 0000000000661598, Data: 31
Address: 00000000006615b0, Data: 2
Address: 00000000006615c8, Data: 123

List setelah pengurutan:
Address: 00000000006615b0, Data: 2
Address: 00000000006615f0, Data: 3
Address: 0000000000661598, Data: 31
Address: 00000000006615c8, Data: 123
PS C:\Users\TOSHIBA\Downloads\ASD Praktikum\Materi\Tugas Baru>
```