TUGAS OTH

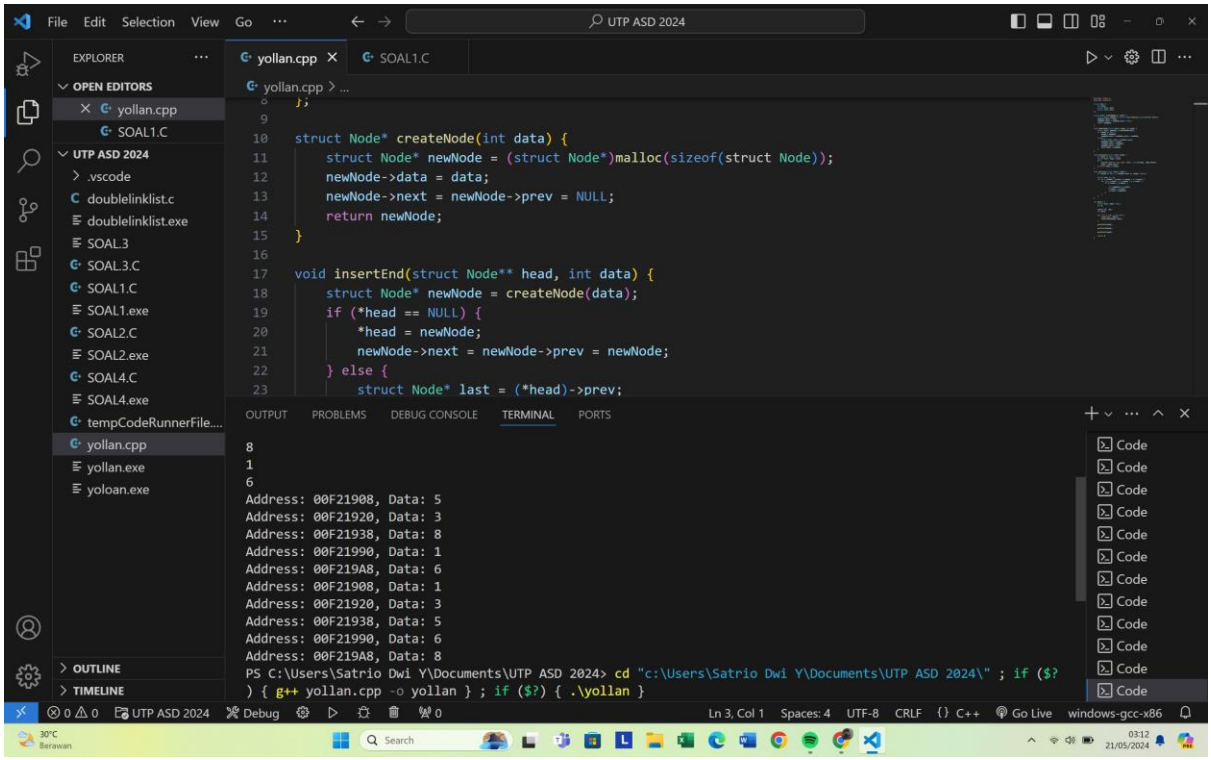
SATRIO DWI YANDA ARIFIN

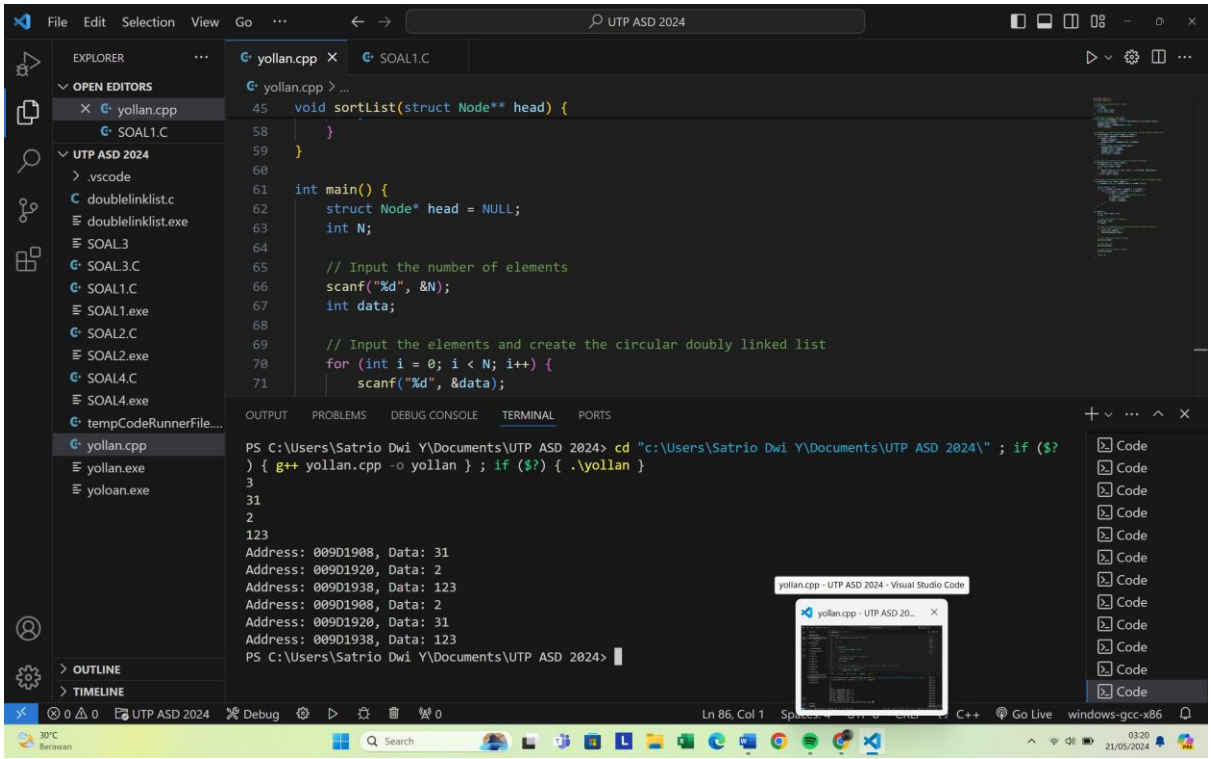
1203230020

IF 03-01

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

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

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

void insertEnd(struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
        newNode->next = newNode->prev = newNode;
    } else {
        struct Node* last = (*head)->prev;
        newNode->next = *head;
        (*head)->prev = newNode;
        newNode->prev = last;
        last->next = newNode;
    }
}

void printList(struct Node* head) {
    if (head == NULL) return;
    struct Node* temp = head;
    do {
        printf("Address: %p, Data: %d\n", (void*)temp, temp->data);
        temp = temp->next;
    } while (temp != head);
```

```c
}

void sortList(struct Node** head) {
    if (*head == NULL || (*head)->next == *head) return;

    struct Node *i, *j;
    for (i = (*head); i->next != *head; i = i->next) {
        for (j = i->next; j != *head; j = j->next) {
            if (i->data > j->data) {

                int tempData = i->data;
                i->data = j->data;
                j->data = tempData;
            }
        }
    }
}

int main() {
    struct Node* head = NULL;
    int N;

    scanf("%d", &N);
    int data;

    for (int i = 0; i < N; i++) {
        scanf("%d", &data);
        insertEnd(&head, data);
    }

    printList(head);

    sortList(&head);

    printList(head);

    return 0;
}
```





Struktur Node dan Fungsi-fungsinya

Definisi Struktur Node

c
Copy code
typedef struct Node int struct Node next struct Node prev

- **`struct Node`** mendefinisikan struktur dari sebuah node dalam linked list.
- **`data`** menyimpan nilai integer dari node.
- **`next`** adalah pointer yang menunjuk ke node berikutnya.
- **`prev`** adalah pointer yang menunjuk ke node sebelumnya.

Fungsi createNode

c
Copy code
createNode int malloc sizeof return

- **`createNode`** membuat node baru dengan nilai **`data`**.
- **`malloc`** mengalokasikan memori untuk node baru.
- **`next`** dan **`prev`** dari node baru menunjuk ke dirinya sendiri (circular).

Fungsi insertEnd

c
Copy code
insertEnd int if return return

- **`insertEnd`** menambahkan node baru di akhir linked list.
- Jika **`head`** adalah **`NULL`**, list kosong, node baru menjadi head.
- Jika list tidak kosong, node baru ditambahkan di antara node tail dan head.
- **`tail`** adalah node terakhir sebelum node baru.
- **`newNode`** menjadi node terakhir dan menghubungkan dirinya dengan **`tail`** dan **`head`**.

Fungsi printList

c
Copy code
void printList if return do printf "Address: %p, Data: %d\n" void while

- **`printList`** mencetak alamat dan data dari setiap node dalam list.
- Jika **`head`** adalah **`NULL`**, list kosong dan tidak ada yang dicetak.
- Loop **`do-while`** digunakan untuk mengunjungi setiap node mulai dari **`head`** hingga kembali ke **`head`**.

Fungsi sortList

```c
Copy code
sortList if return NULL NULL do
if else
while if
else
while return
```

- **`sortList`** mengurutkan linked list dengan mengubah posisi node, bukan data.
- **`sorted`** adalah pointer untuk menjaga list yang sudah diurutkan.
- **`curr`** adalah node saat ini yang sedang diurutkan.
- Loop **`do-while`** berjalan sampai semua node diproses.
- **`next`** menyimpan node berikutnya sebelum **`curr`** dilepaskan dari list asalnya.
- Node **`curr`** di-insert pada posisi yang tepat dalam list **`sorted`**.
- Jika **`sorted`** masih kosong, **`curr`** menjadi node pertama **`sorted`**.
- Node **`curr`** ditempatkan sebelum node **`temp`** jika data **`curr`** lebih kecil dari **`temp`**.
- Fungsi memastikan bahwa node-node di **`sorted`** tetap terurut.

Fungsi main

```c
Copy code
int main int scanf "%d" NULL for int 0 int
scanf "%d" printf "List before sorting:\n"
printf "List after sorting:\n" return 0
```

- Fungsi **`main`** adalah titik awal program.
- **`N`** adalah jumlah node yang akan dimasukkan.
- Loop **`for`** membaca **`N`** data dan memasukkan ke linked list menggunakan **`insertEnd`**.
- Mencetak list sebelum pengurutan.
- Mengurutkan list menggunakan **`sortList`**.
- Mencetak list setelah pengurutan.

Dengan penjelasan ini, setiap bagian kode memiliki fungsinya yang jelas untuk membuat, menampilkan, dan mengurutkan circular double linked list.