**LAPORAN PRATIKUM STRUKTUR DATA**

**JOBSHEET 5**



**Disusun Oleh :**

**NAJWA ELDIARA OWILIA TIKSA**

**BP/NIM : 2023/23343079**

**Dosen Pengampu :**

**Randi Proska Sandra, S.Pd, M.Sc**

**Kode Kelas : 202323430157**

**PROGRAM STUDI S1 INFORMATIKA**

**DEPARTEMEN ELEKTRONIKA**

**FAKULITAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2023**

**1.**

1. **Source Code**

//created by 23343079\_Najwa Eldiara Owilia Tiksa

#include<stdio.h>

#include<stdlib.h>

struct node {

int no\_pol;

struct node\* next;

};

void display(struct node\* head) {

struct node \*temp = head;

printf("\n\nUrutan parkiran: \n");

while(temp != NULL) {

printf("%d ---> ", temp->no\_pol);

temp = temp->next;

}

}

void insertAtFront(struct node\*\* headRef, int value) {

struct node\* head = \*headRef;

struct node\* newNode;

newNode = (struct node\*)malloc(sizeof(struct node));

newNode->no\_pol = value;

newNode->next = head;

head = newNode;

\*headRef = head;

}

int main() {

struct node \*head = NULL;

struct node \*avanza = NULL;

struct node \*hilux = NULL;

struct node \*kijangF1 = NULL;

avanza = (struct node\*)malloc(sizeof(struct node));

hilux = (struct node\*)malloc(sizeof(struct node));

kijangF1 = (struct node\*)malloc(sizeof(struct node));

avanza->no\_pol = 1234;

hilux->no\_pol = 2345;

kijangF1->no\_pol = 3456;

/\* Connect nodes \*/

avanza->next = hilux;

hilux->next = kijangF1;

kijangF1->next = NULL;

/\* Save address of first node in head \*/

head = avanza;

insertAtFront(&head, 46543);

display(head);

return 0;

}

1. **Penjelasan Program**

Program di atas adalah implementasi dari linked list sederhana dalam bahasa C. Program ini menciptakan sebuah linked list yang merepresentasikan urutan parkiran mobil berdasarkan nomor polisi. Berikut adalah penjelasan lebih detail mengenai bagian-bagian dari program tersebut:

Algoritma yang Digunakan

Program ini menggunakan algoritma linked list yang mendukung operasi dasar sebagai berikut:

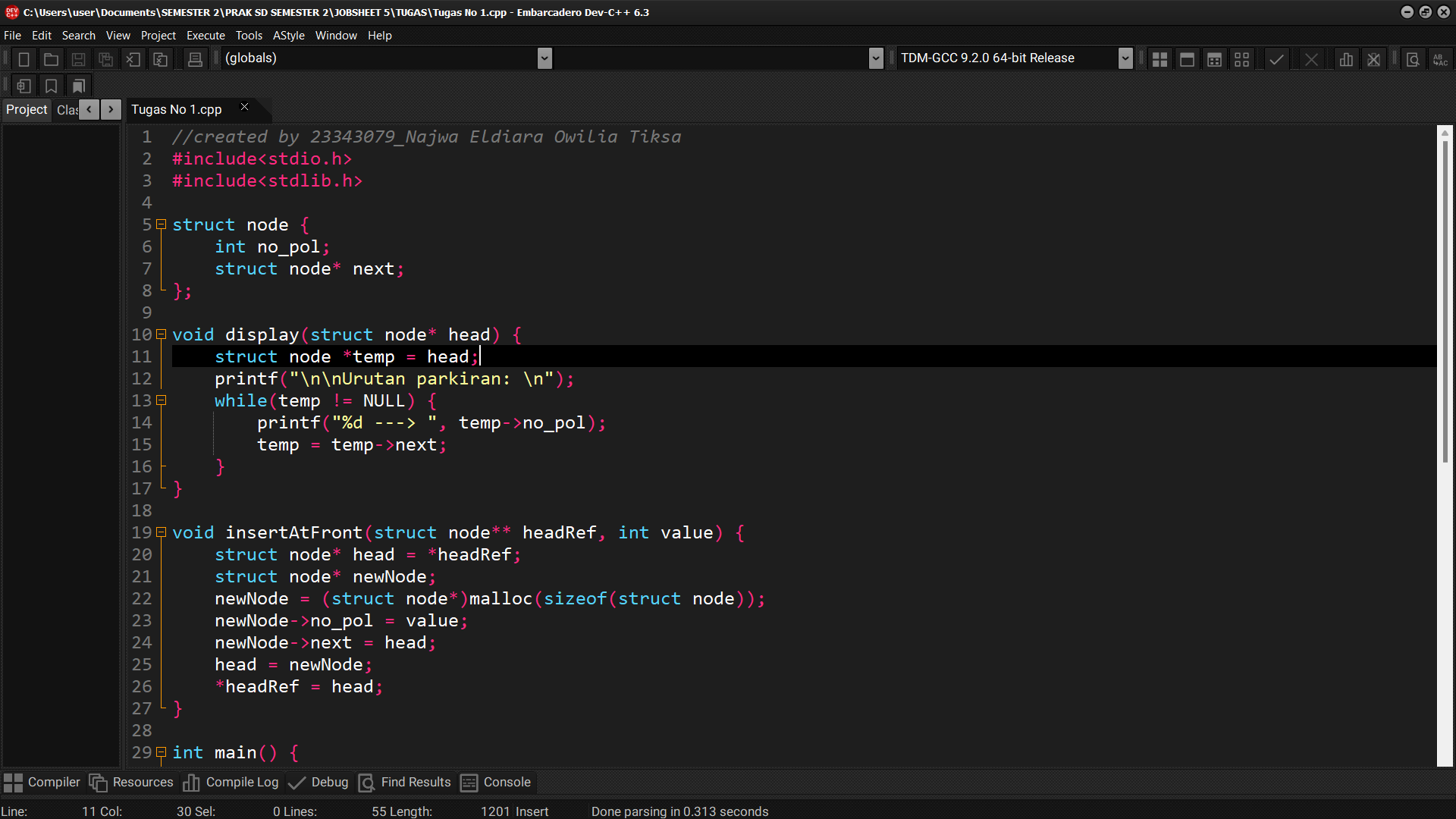
- Inisialisasi: Membuat node dan mengatur nilai awal.

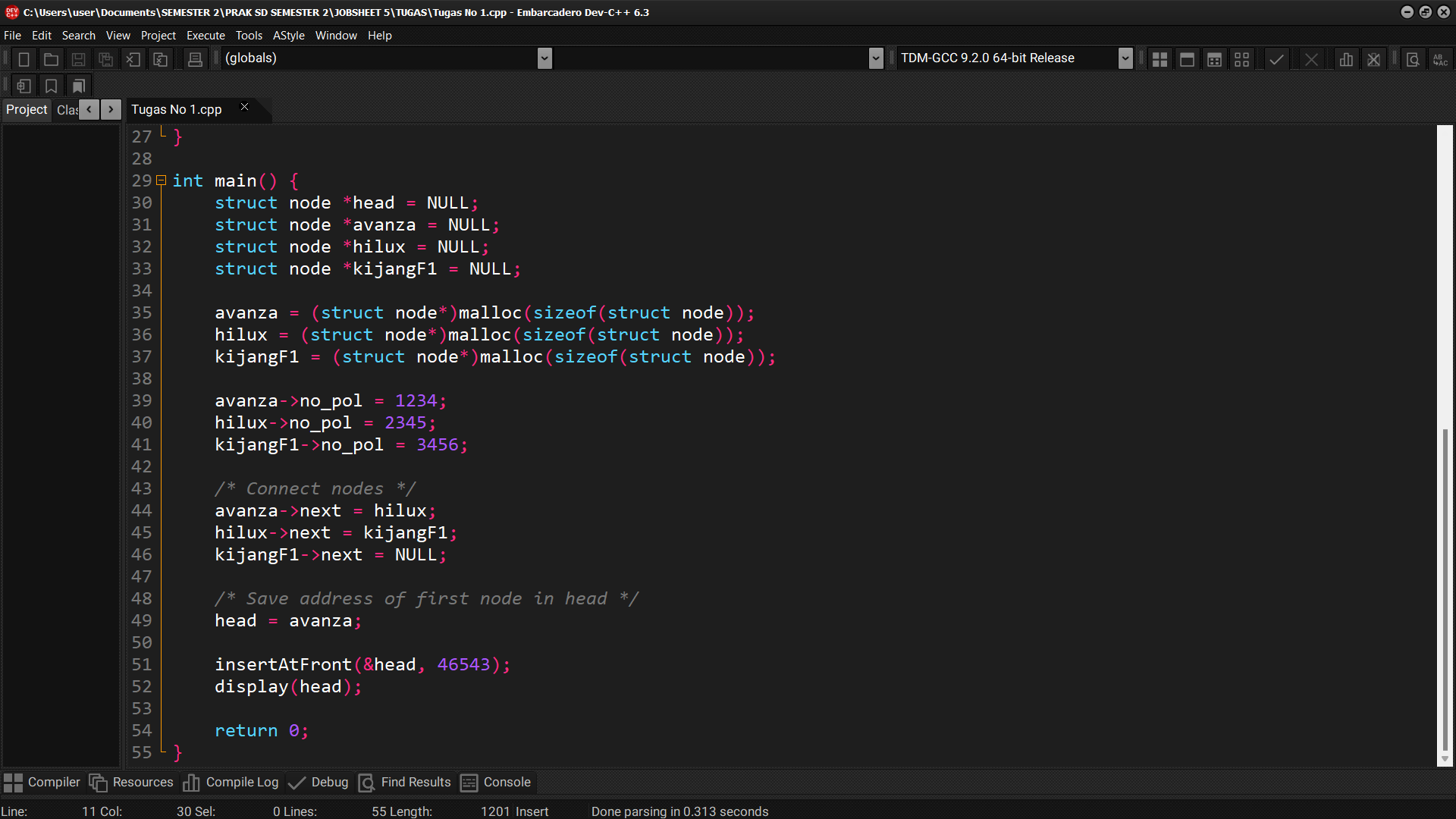
- Penyisipan di Depan: Menambahkan node baru di awal linked list.

- Traversal: Menelusuri semua node dalam linked list untuk menampilkan data.

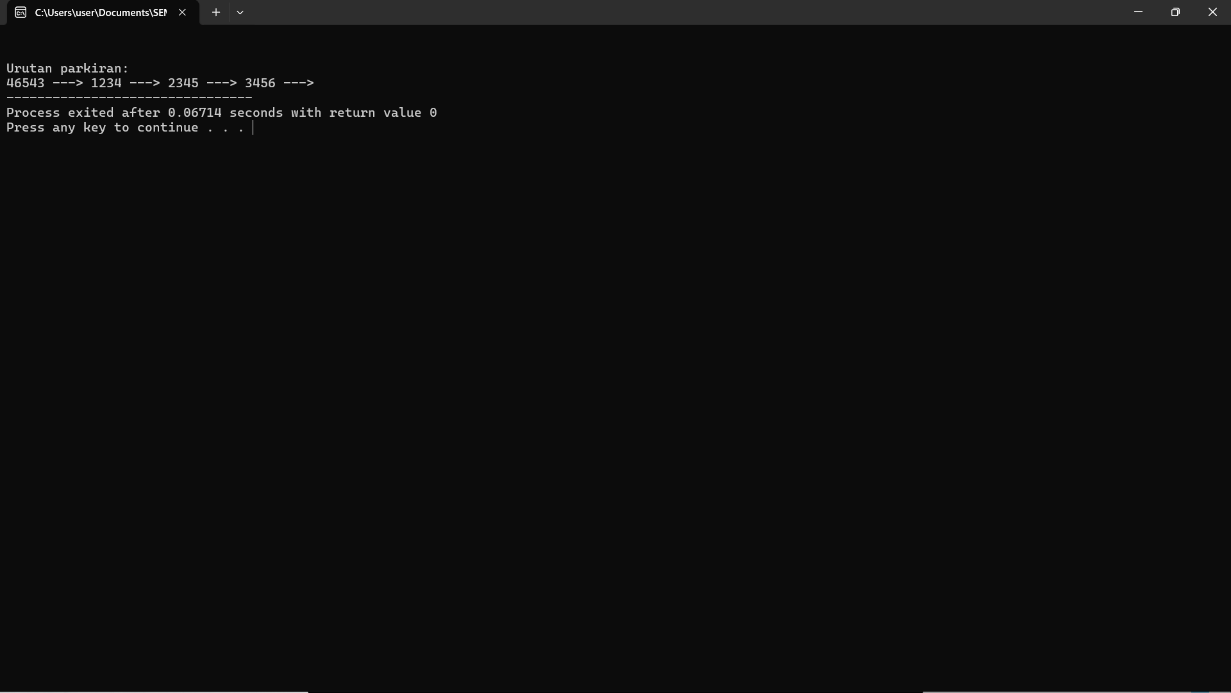
Operasi penyisipan di depan memiliki kompleksitas waktu O(1) karena hanya melibatkan perubahan pointer. Traversal untuk menampilkan data memiliki kompleksitas O(n), di mana n adalah jumlah node dalam linked list.

1. **Screenshot program**





1. **Output**



**2.**

1. **Source Code**

//created by 23343079\_Najwa Eldiara Owilia Tiksa

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct gudang {

char nama\_barang[50];

int harga;

struct gudang \*next;

struct gudang \*prev;

};

void cetak(struct gudang\* head) {

struct gudang \*temp = head;

printf("\n\nList elements are - \n");

while(temp != NULL) {

printf("%s = %d ---> ", temp->nama\_barang, temp->harga);

temp = temp->next;

}

}

int main() {

struct gudang \*head = NULL;

struct gudang \*one = NULL;

struct gudang \*two = NULL;

struct gudang \*three = NULL;

/\* Allocate memory \*/

one = (struct gudang\*)malloc(sizeof(struct gudang));

two = (struct gudang\*)malloc(sizeof(struct gudang));

three = (struct gudang\*)malloc(sizeof(struct gudang));

/\* Assign data values \*/

one->harga = 1000;

strcpy(one->nama\_barang, "obat nyamuk");

two->harga = 5000;

strcpy(two->nama\_barang, "Gula 1 kilo");

three->harga = 50000;

strcpy(three->nama\_barang, "Minyak Goreng");

/\* Connect nodes \*/

one->next = two;

one->prev = NULL;

two->next = three;

two->prev = one;

three->next = NULL;

three->prev = two;

/\* Save address of first node in head \*/

head = one;

cetak(head);

return 0;

}

1. **Penjelasan Program**

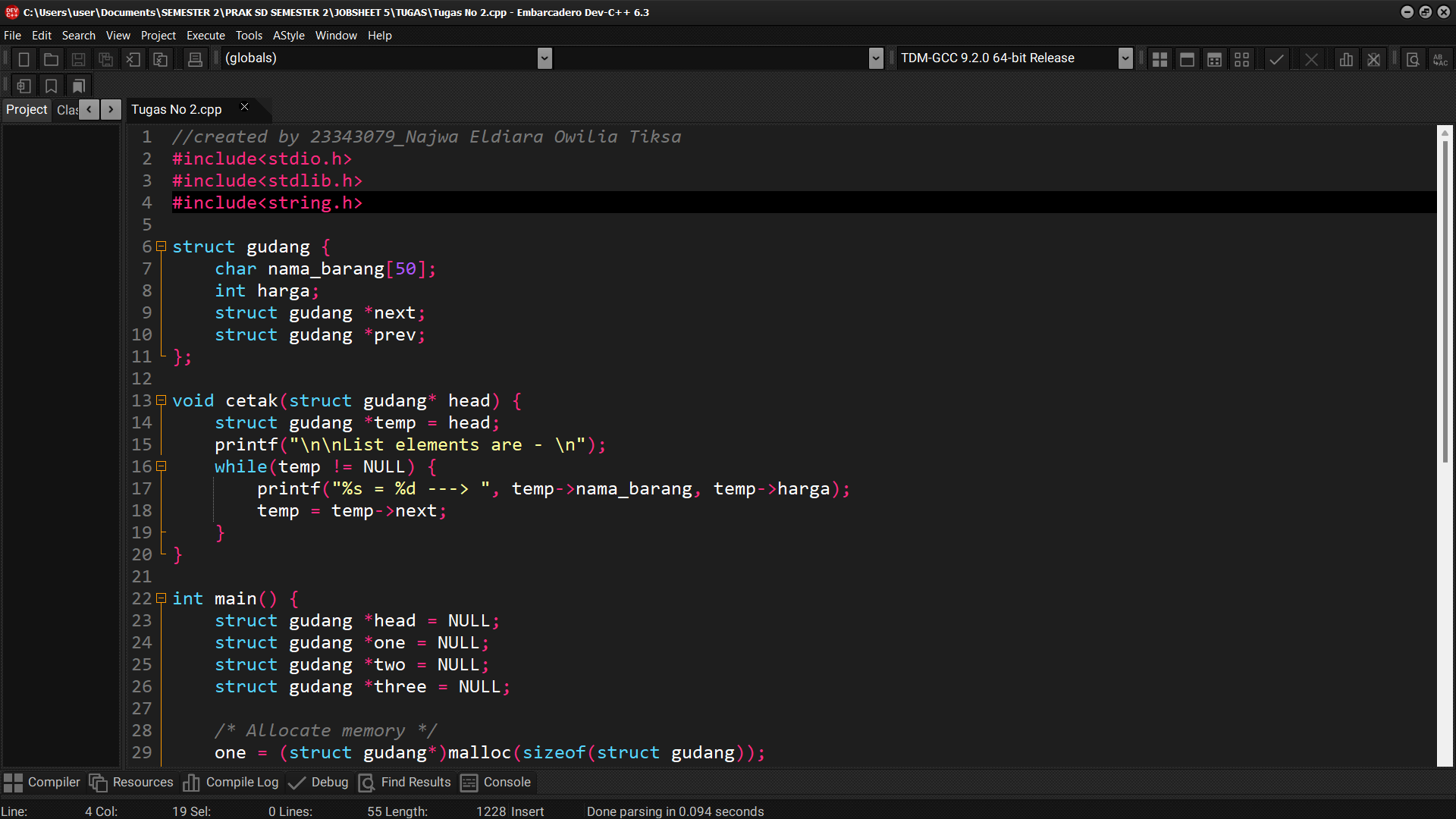
Program berikut ini adalah contoh implementasi linked list ganda (doubly linked list) dalam bahasa C. Linked list adalah struktur data dinamis yang terdiri dari node-node yang terhubung secara berurutan. Dalam linked list ganda, setiap node memiliki dua pointer: satu mengarah ke node berikutnya (next) dan satu lagi mengarah ke node sebelumnya (prev).

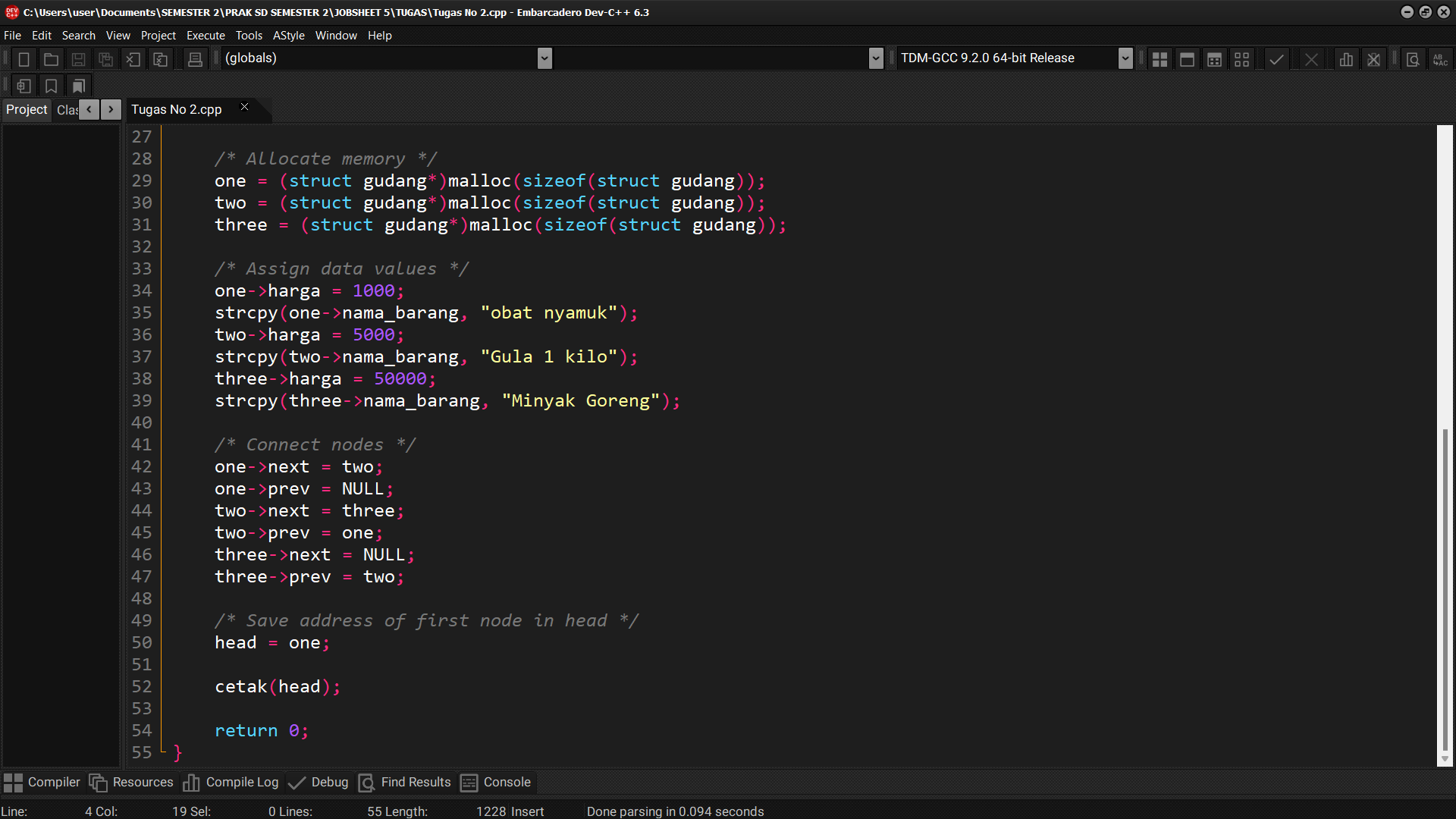
Algoritma yang digunakan

Algoritma utama yang digunakan dalam program ini adalah linked list ganda (doubly linked list), dimana setiap node memiliki pointer untuk node berikutnya dan sebelumnya. Ini memungkinkan traversal dua arah (forward dan backward) dari linked list.

Program ini terutama menunjukkan cara dasar membangun dan mencetak elemen dari linked list ganda, menggunakan operasi alokasi memori, penugasan nilai, dan penghubungan node dalam linked list.

1. **Screenshot program**





1. **Output**

