# UJIAN PRAKTIKUM

Mata Praktikum : Struktur Data

Kelas : 2IA16

Praktikum ke - : 8

Tanggal : Sabtu, 04 Desember 2021

Materi : Ujian

NPM : 50420093

Nama : Ajay Alfredo Almani

Ketua Asisten : Thomas Adya Dewangga

Nama Asisten :

Paraf Asisten :

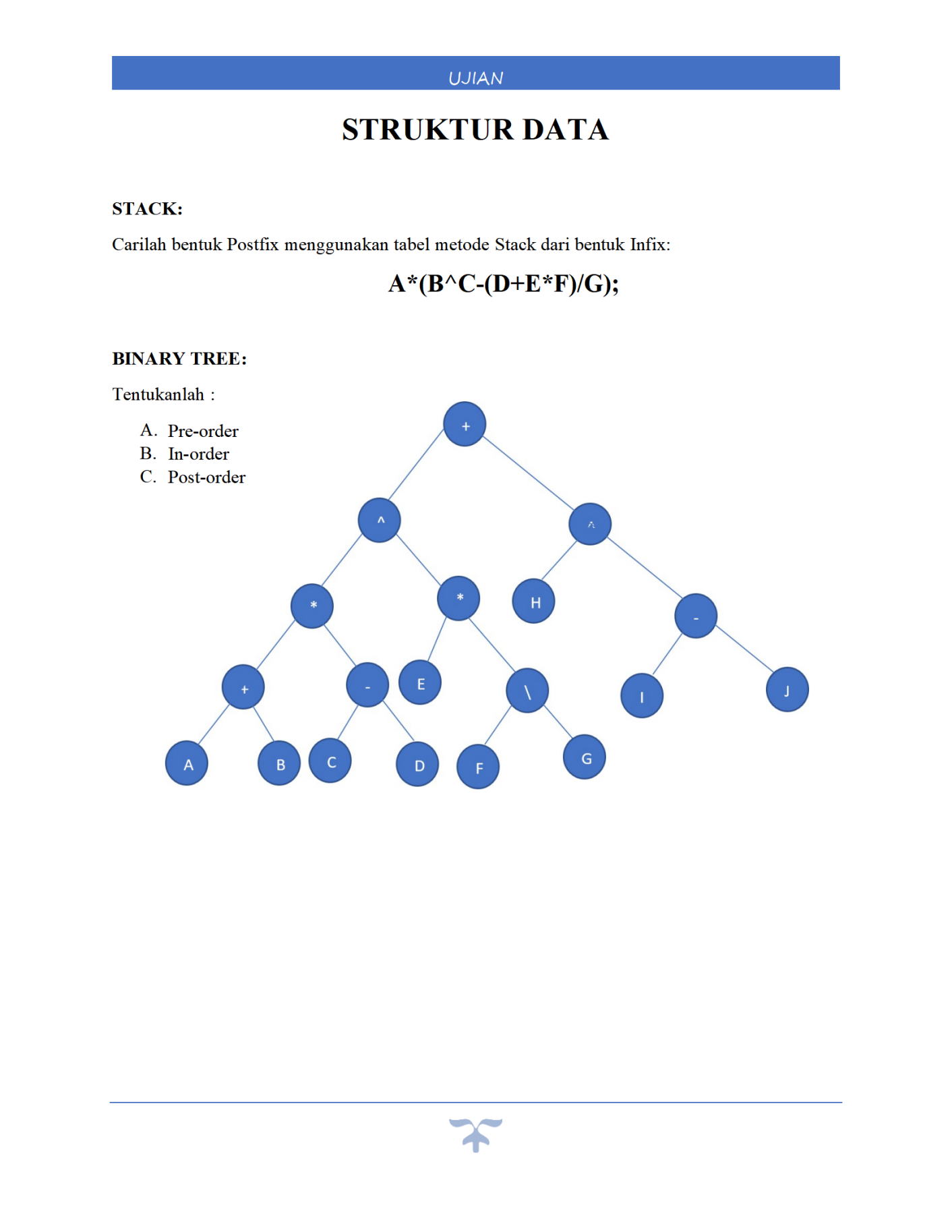
Jumlah Lembar : 9 Lembar



# LABORATORIUM TEKNIK INFORMATIKA UNIVERSITAS GUNADARMA

**2021**

**SOAL UJIAN TEORI**

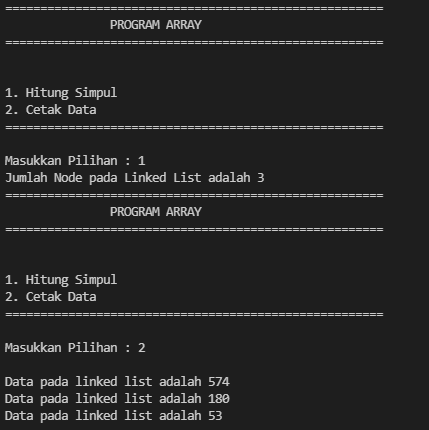
****

**SOAL UJIAN CODE RUNNING**

1. Ujian Coding Struktur Data ini adalah menjalankan program linked list

2. Program ini belum sempurna dan akan menjalankan hasil error.

3. Tugas kalian adalah menyempurnakan code tersebut sampai bisa menghasilkan output NPM seperti pada gambar (hasil ujian\_linkedlist.png)



4. Jelaskan secara singkat algoritma linked list yang berhasil kalian jalankan.

5. Jika program belum berhasil berjalan (masih error/tidak sesuai output), kumpulkan code yang telah kalian kerjakan beserta penjelasan singkat mengenai algoritma linked list.

**JAWABAN UJIAN TEORI**

**Stack**

Notasi Infix : A\*(B^C-(D+E\*F)/G)

|  |  |  |
| --- | --- | --- |
| **Input String** | **Output Stack** | **Operator Stack** |
| A\*(B^C-(D+E\*F)/G) | A |  |
| A\*(B^C-(D+E\*F)/G) | A | \* |
| A\*(B^C-(D+E\*F)/G) | A | \*( |
| A\*(B^C-(D+E\*F)/G) | AB | \*( |
| A\*(B^C-(D+E\*F)/G) | AB | \*(^ |
| A\*(B^C-(D+E\*F)/G) | ABC | \*(^ |
| A\*(B^C-(D+E\*F)/G) | ABC^ | \*(- |
| A\*(B^C-(D+E\*F)/G) | ABC^ | \*(-( |
| A\*(B^C-(D+E\*F)/G) | ABC^D | \*(-( |
| A\*(B^C-(D+E\*F)/G) | ABC^D | \*(-(+ |
| A\*(B^C-(D+E\*F)/G) | ABC^DE | \*(-(+ |
| A\*(B^C-(D+E\*F)/G) | ABC^DE | \*(-(+\* |
| A\*(B^C-(D+E\*F)/G) | ABC^DEF | \*(-(+\* |
| A\*(B^C-(D+E\*F)/G) | ABC^DEF\*+ | \*(- |
| A\*(B^C-(D+E\*F)/G) | ABC^DEF\*+ | \*(-/ |
| A\*(B^C-(D+E\*F)/G) | ABC^DEF\*+G | \*(-/ |
| A\*(B^C-(D+E\*F)/G) | ABC^DEF\*+G/-\* |  |

**BINARY TREE**

+

^

^

H

-

I

J

\*

\*

E

/

F

G

-

C

D

+

A

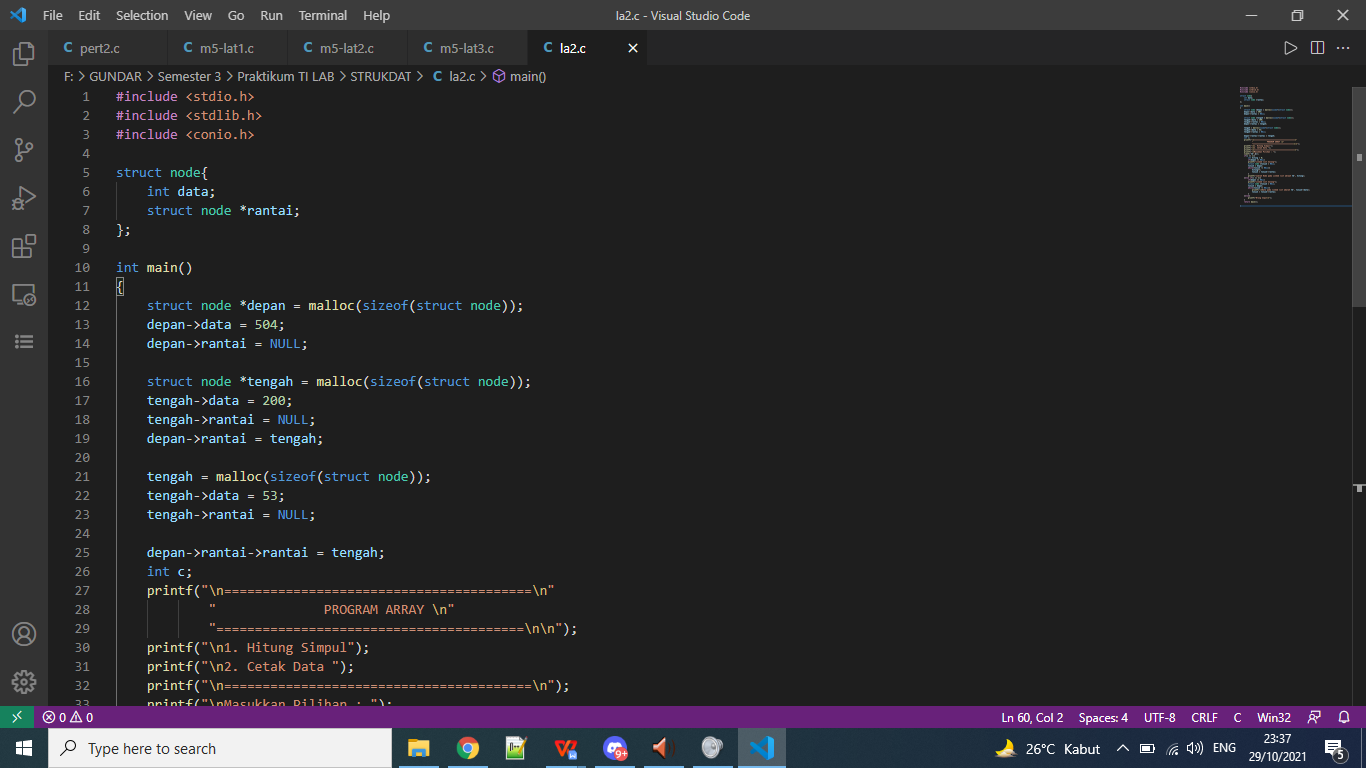
B

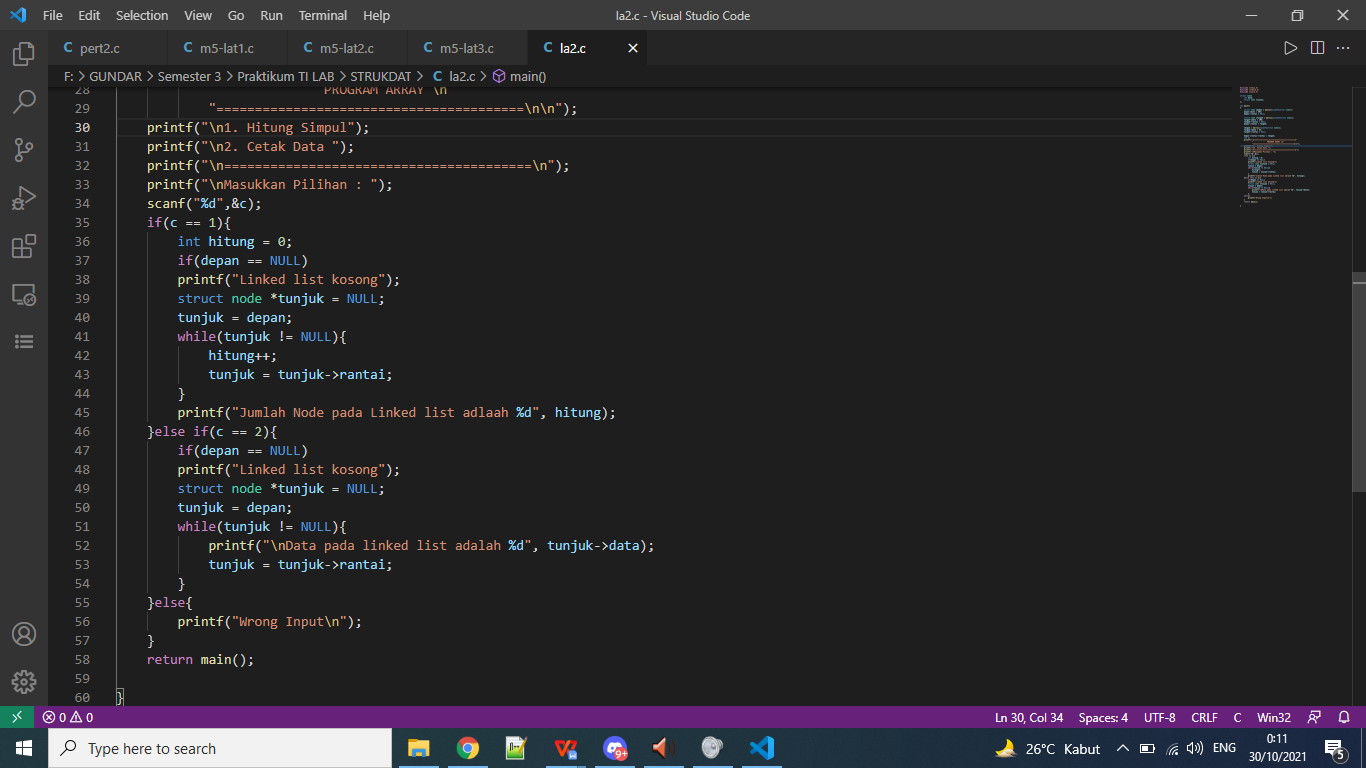
Tentukanlah :

1. Pre-order : + ^ \* + A B – C D \* E / F G ^ H – I J
2. In-order : A + B \* C – D ^ E \* F / G + H ^ I – J
3. Post-order : A B + C D – \* E F G / \* ^ H I J – ^ +

**JAWABAN UJIAN CODE RUNNING**

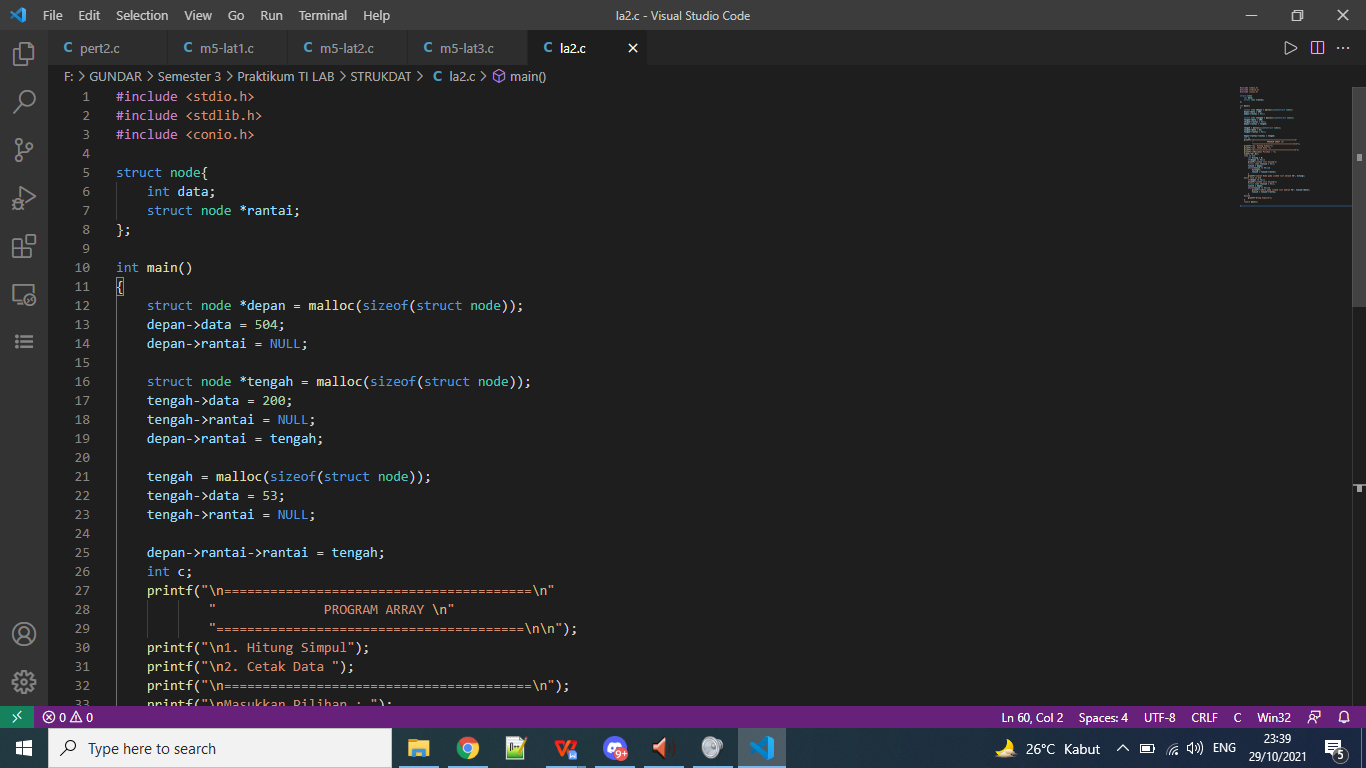
**LISTING PROGRAM**



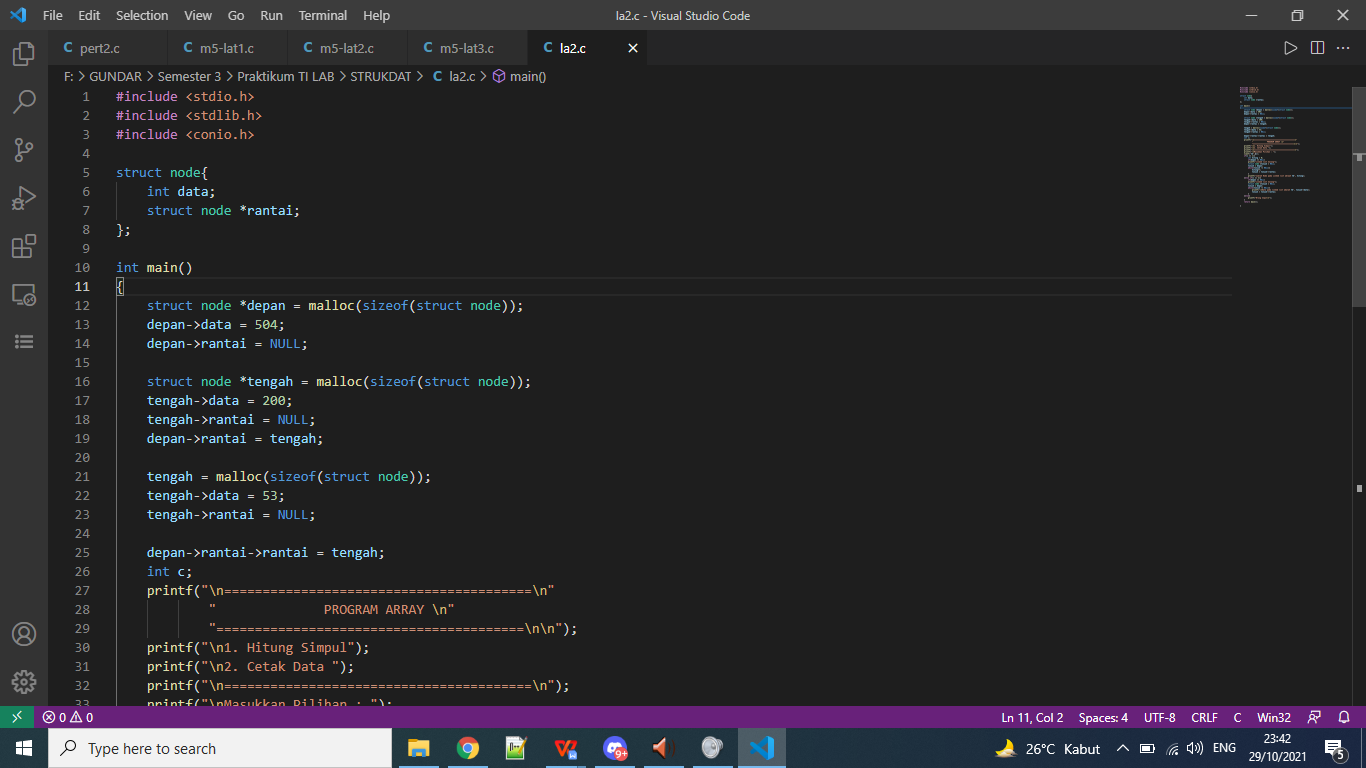


**LOGIKA PROGRAM**

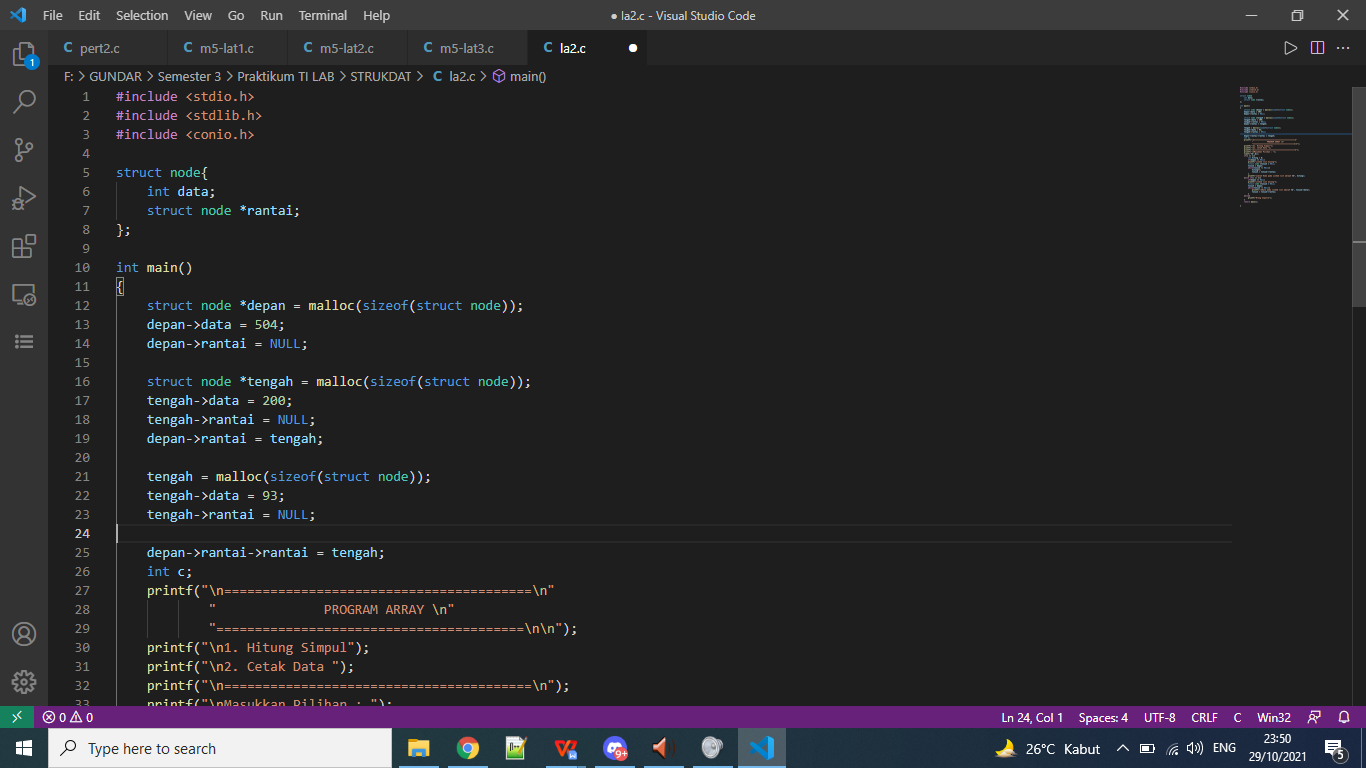
Pada Ujian Praktikum ini yaitu Struktur Data, Saya mendapat materi Linked List. Untuk membuat program sederhana dari bahasa C. Sebelumnya kita akan membahas C singkat. Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Meskipun C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. Dibawah ini saya membuat Program Linked List dari NPM. Berikut Langkah Langkah dibawah ini untuk membuat Program Linked List dari NPM dari bahasa C yang bisa anda di lihat dibawah ini.



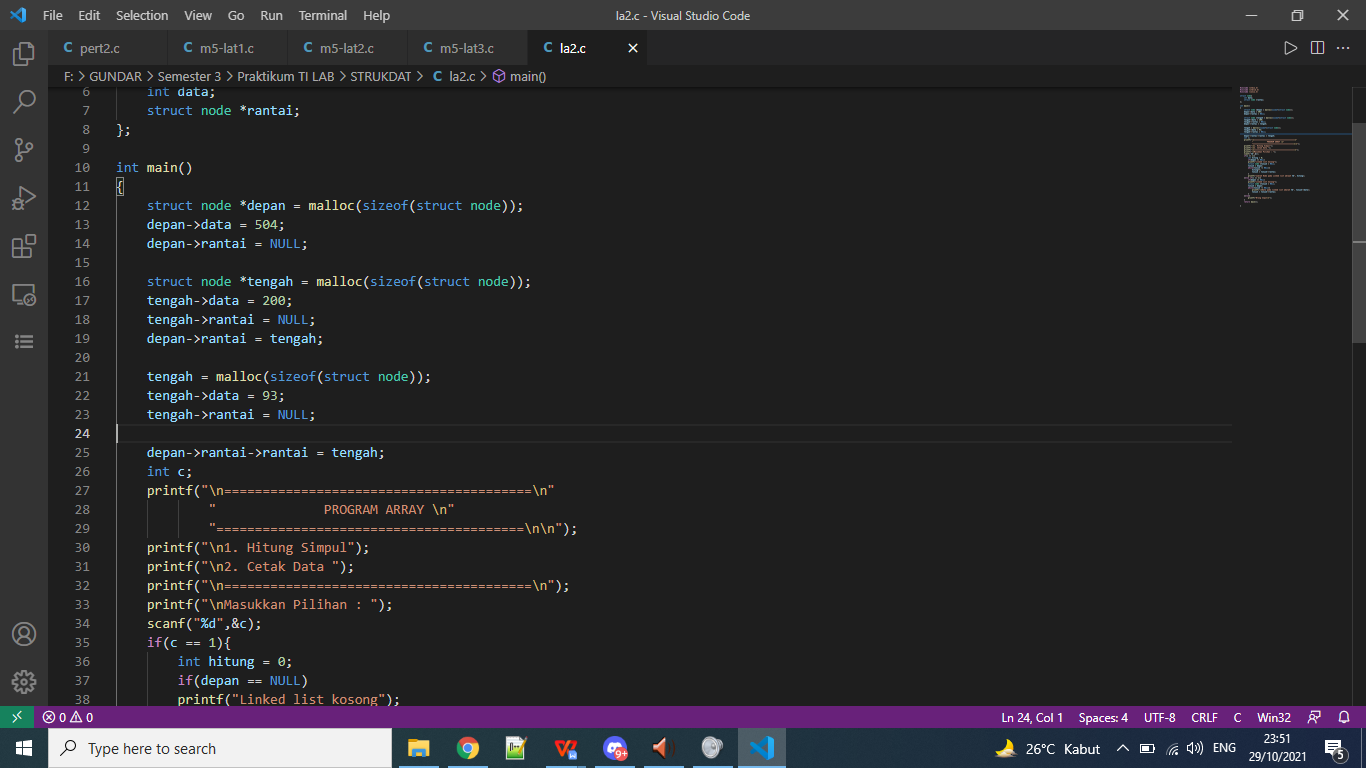
Langkah pertama kita harus memiliki text editor untuk menjalankan Bahasa C. yang saya gunakan disini ialah Aplikasi Visual Studio Code. Agar Bahasa C bisa di jalankan di Visual Studio Code harus meng install beberapa ekstensi terlebih dahulu dan saya menginstall juga MinGW. Yang bisa di download disini ( <https://nuwen.net/mingw.html> ). Selanjutnya jika sudah bisa menjalankan Bahasa C di dalam Visual Studio Code. Selanjutnya setelah meng install MinGW jangan lupa mengatur Environment nya. Lalu Tidak lupa meng install Ekstensi Bahasa C dan Code Runner di Visual Studio Code. Anda bisa mengikuti code seperti gambar di atas. Dan saya akan menjelaskan tiap baris dalam code tersebut. Untuk baris pertama yaitu **#include <stdio.h>** Yang berguna untuk memasukkan sebuah file khusus yang memungkinkan kita mengakses berbagai fitur tambahan dalam bahasa C. Lalu kita masuk ke baris kedua yaitu **#include <stdlib.h>** Yang Merupakan file header yang berfungsi untuk operasi pembanding dan operasi konversi.. Lalu kita masuk ke baris ketiga **#include <conio.h>** Yang merupakan file header yang berfungsi untuk menampilkan hasil antarmuka kepada pengguna.



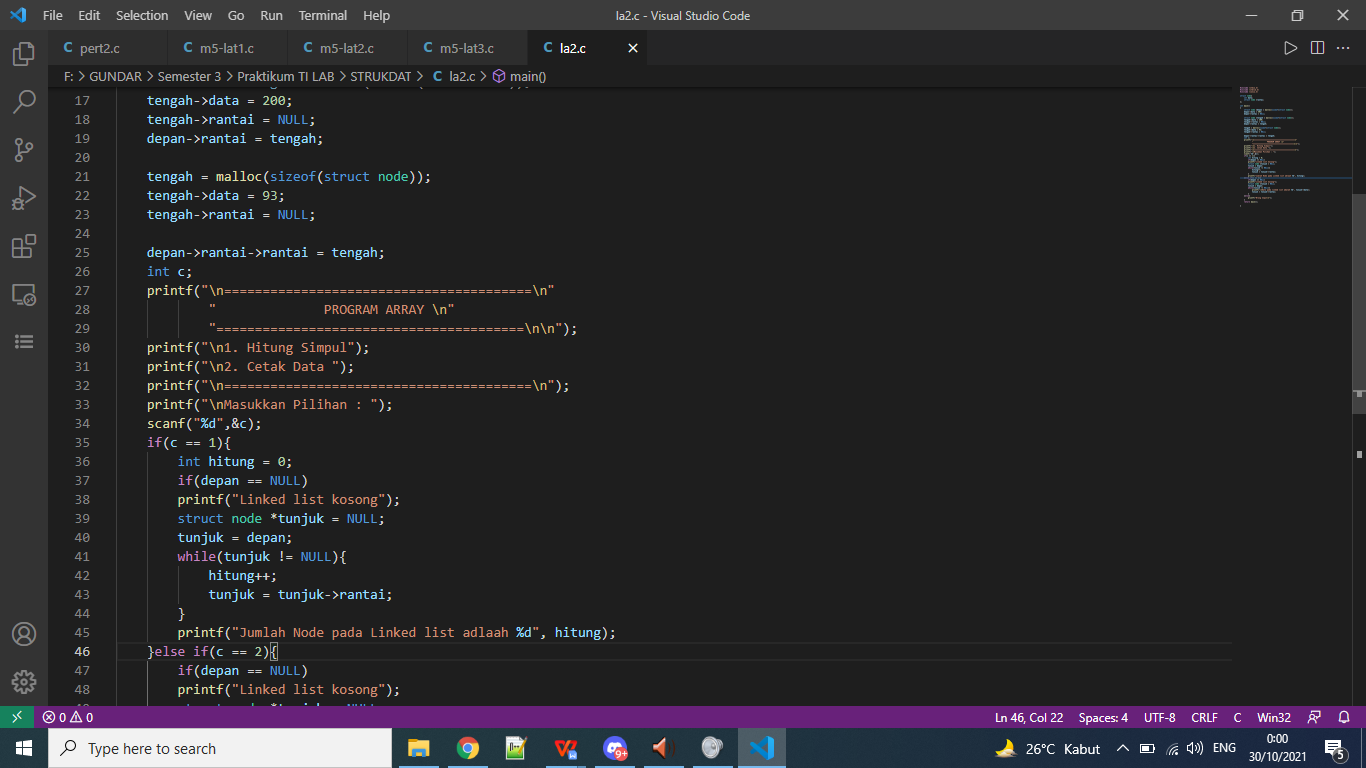
Langkah Selanjutnya adalah seperti gambar di atas pada baris ke lima yaitu berfungsi untuk pembentukan **pointer utama**. dan juga sebuah struktur dengan nama node lalu selanjutnya **int data;** yang sebagai Variable Interger lalu ada **struct node \*rantai;** sebagai **pointer data.**



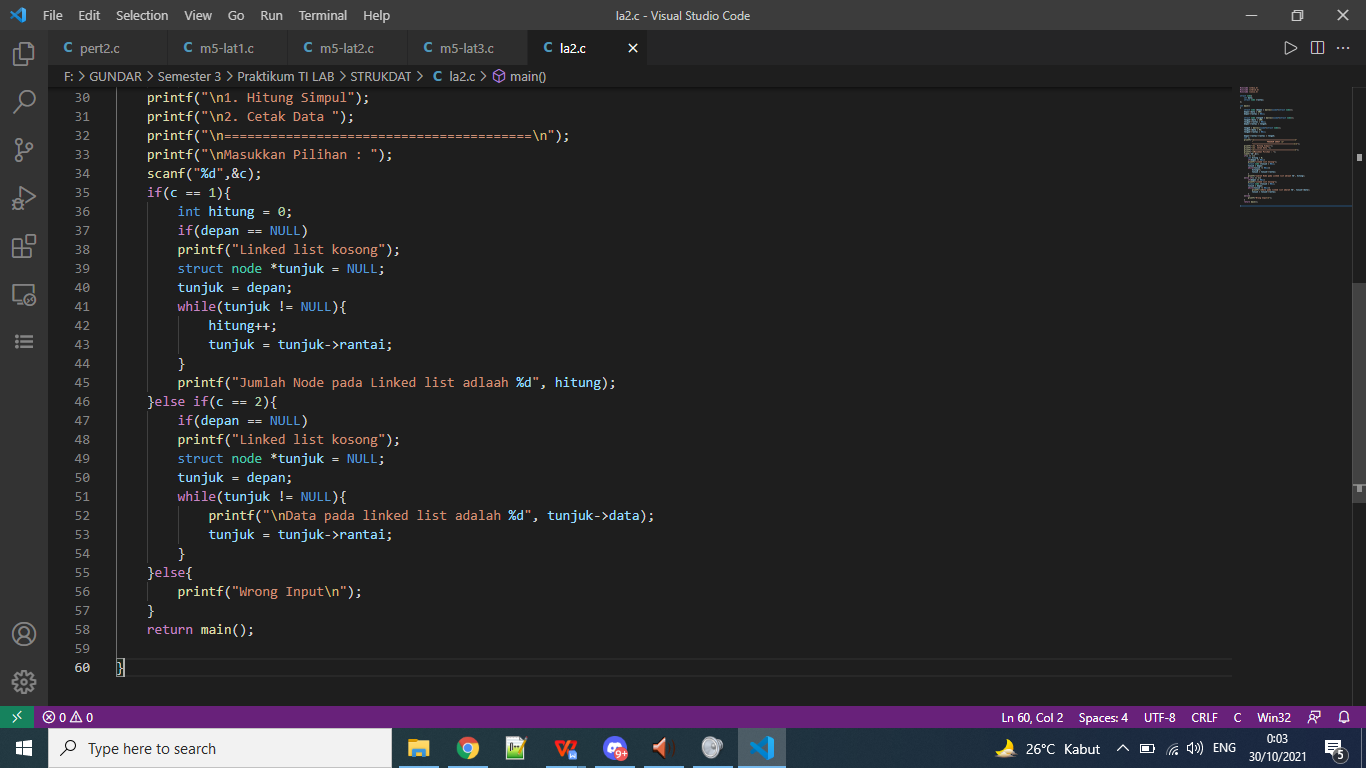
Langkah Selanjutnya adalah seperti gambar di atas pada baris ke 10 yaitu  **int main(),** adalah **fungsi (function**) utama, fungsi yang akan dibaca oleh kompilator pertama kali secara otomatis, tidak akan ada function yang otomatis dibaca oleh kompilator kecuali fungsi utama int main(), fungsi utama bagaikan sebuah kepala dari sebuah program yang mengatur arah kompilator. int dalam fungsi utama adalah sebuah return type integer. Dalam peraturan C++ fungsi utama memang harus menggunakan return type integer. **struct node \*depan = malloc(sizeof(struct node));** yang berfungsi sebagai Struct node depan adalah variable pertama. **depan->data = (504)**; yang berisi variable data **504**. **struct node \*tengah = malloc(sizeof(struct node)); tengah->data = 200;** yang berisi Struct node tengah adalah pointer / variable yg berisi data **200**. Selanjutnya ada **tengah = malloc(sizeof(struct node)); tengah->data = 93;** yang berisi Struct node tengah adalah pointer / variable yg berisi data **93**.



Langkah Selanjutnya adalah seperti gambar di atas pada baris ke 25 yaitu **depan->rantai->rantai = tengah; hitung\_simpul(depan);** yang berisi metode fungsi simpul yg berhubungan simpul depan. Selanjutnya **int c;** yaitu sebagai fungsi c. Selanjutnya membuat **printf("\n========================================\n" " PROGRAM ARRAY \n" "========================================\n\n");** untuk melakukan pencetakan dan \n ini agar pencetakan output terdapat dalam baris baru atau new line dan mencetak kata **PROGRAM ARAY**. Selanjutnya mencetak 1. Hitung Simpul & 2. Cetak Data dan Memasukkn Pilihan. Lalu **%d** yaitu untuk menampilkan output angka berupa decimal dan **%c** yaitu untuk menampilkan output dengan karakter.



Langkah Selanjutnya adalah seperti gambar di atas pada baris ke 35 yaitu untuk menghitung simpul yang ada dengan percabangan. Kondisi pertama yaitu **if(c == 1){** percabangan pertama yaitu pemilihan angka 1. Deklarasikan variable hitung dengan nilai 0 dengan program **int hitung = 0;** , lalu kita buat percabangan jika depan sama dengan **NULL** maka **“Linked list kosong”** dengan program **if(depan == NULL) printf("Linked list kosong");.** Lalu buat pointer baru dengan nilai **NULL** dengan program **struct node \*tunjuk = NULL;** dan buat pointer **tunjuk = depan.** Setelah itu buat percabangan jika selama kondisi tunjuk tidak sama dengan **NULL** maka hitung akan increment atau bertambah dengan program **while(tunjuk != NULL){ hitung++** , lalu buat pointer tunjuk sama dengan pointer tunjuk ke pointer rantai dengan program **tunjuk = tunjuk->rantai;** .



Selanjutnya adalah pada gambar terakhir yang di atas ini berisikan program terakhir dan juga masuk ke percabangan kondisi kedua dan ketiga. Pertama buat program untuk menghitung simpul yang ada dengan percabangan. Kondisi pertama yaitu **if(c == 2){** sebagai percabangan pertama yaitu pemilihan angka 2, lalu kita buat percabangan jika depan sama dengan **NULL** maka **“Linked list kosong”** dengan prgoram **if(depan == NULL) printf("Linked list kosong");** . Lalu buat pointer baru dengan nilai **NULL** dengan program **struct node \*tunjuk = NULL;** dan buat pointer **tunjuk = depan**, setelah itu buat percabangan jika selama kondisi tunjuk tidak sama dengan **NULL** maka akan tercetak **“Data pada linked list adalah”** dengan program **printf("\nData pada linked list adalah %d", tunjuk->data);** dan berhenti dengan program **tunjuk = tunjuk->rantai;** . Lalu percabangan terakhir jika inputan angka diluar angka 1 dan 2 maka akan tercetak **“Wrong Input”** dengan program **else{ printf("Wrong Input\n");** , dan terakhir untuk mengakhiri program dengan **return main();**

**OUTPUT PROGRAM**

