# LAPORAN AKHIR

Mata Praktikum : Struktur Data

Kelas : 2IA16

Praktikum ke - : 3

Tanggal : Sabtu, 30 Oktober 2021

Materi : Double Linked List

NPM : 50420093

Nama : Ajay Alfredo Almani

Ketua Asisten : Thomas Adya Dewangga

Nama Asisten :

Paraf Asisten :

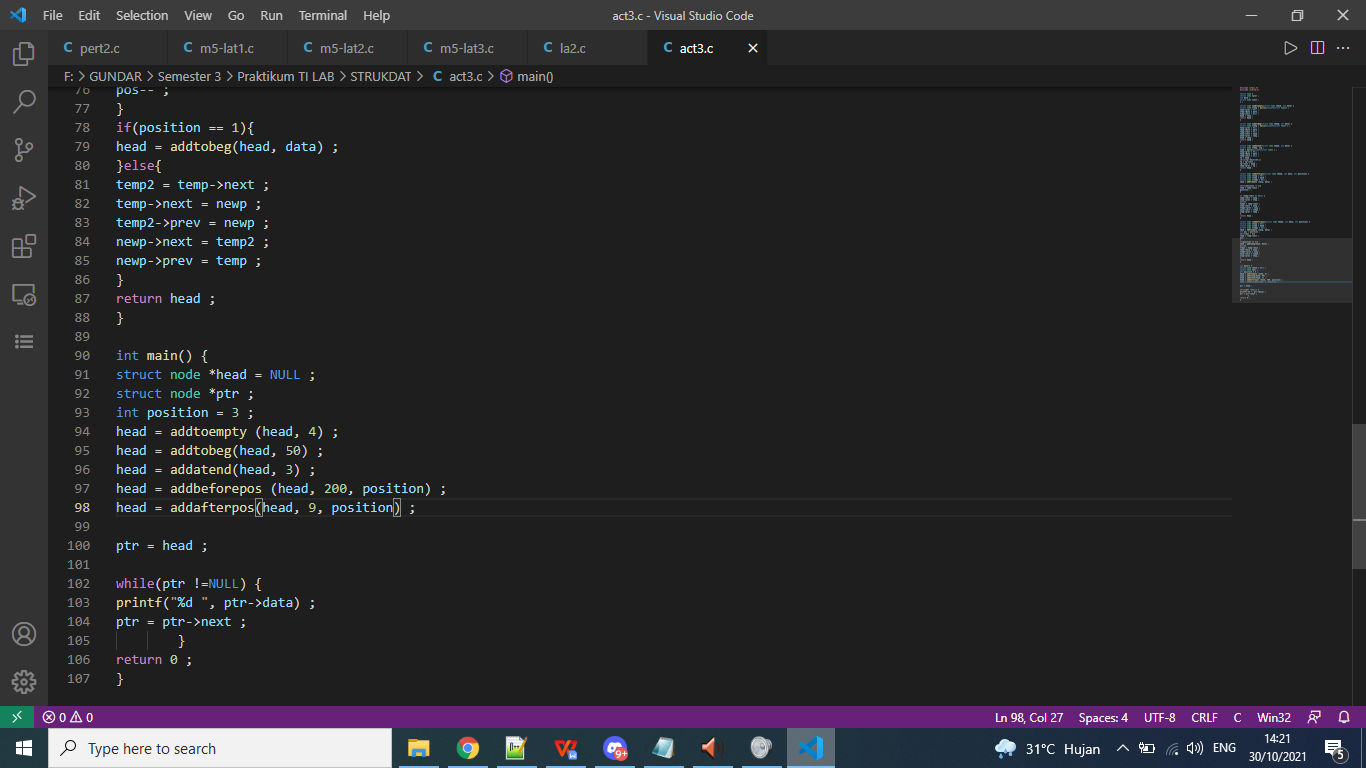
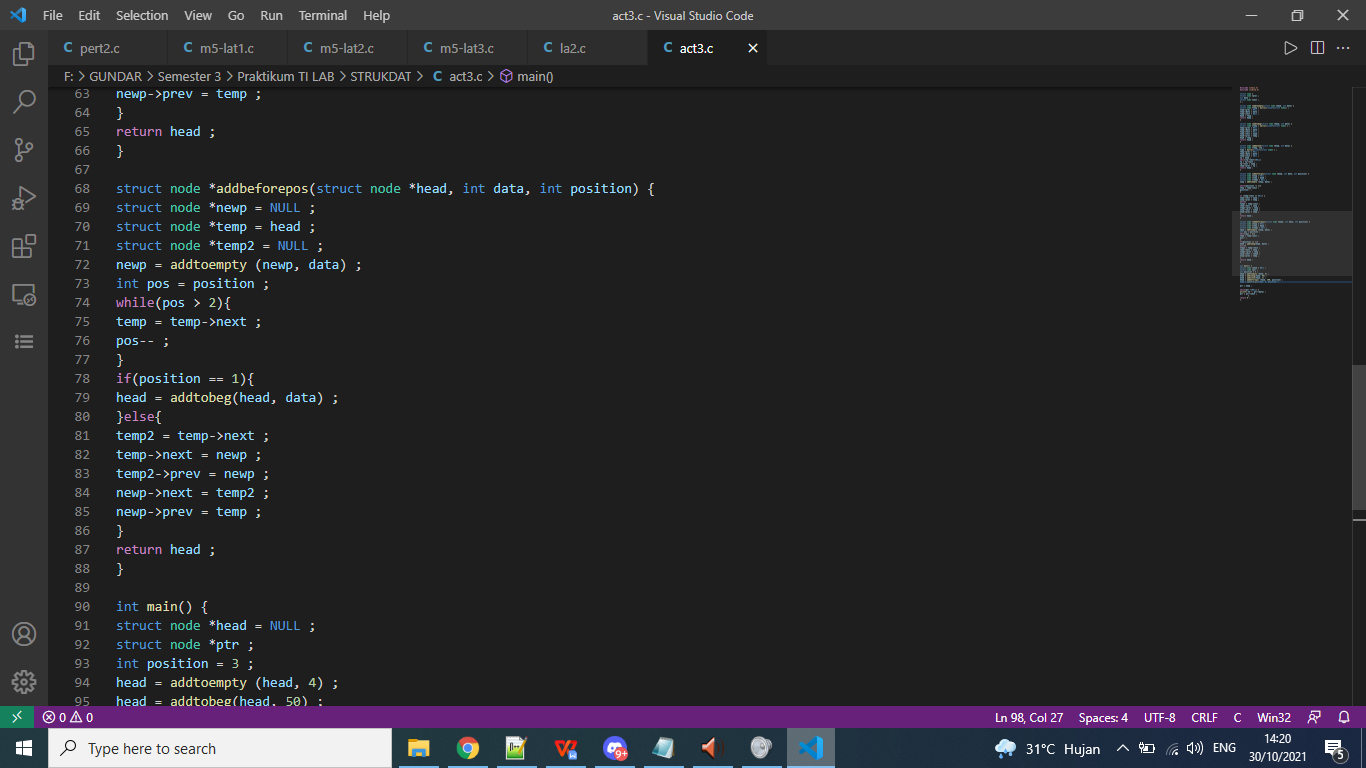
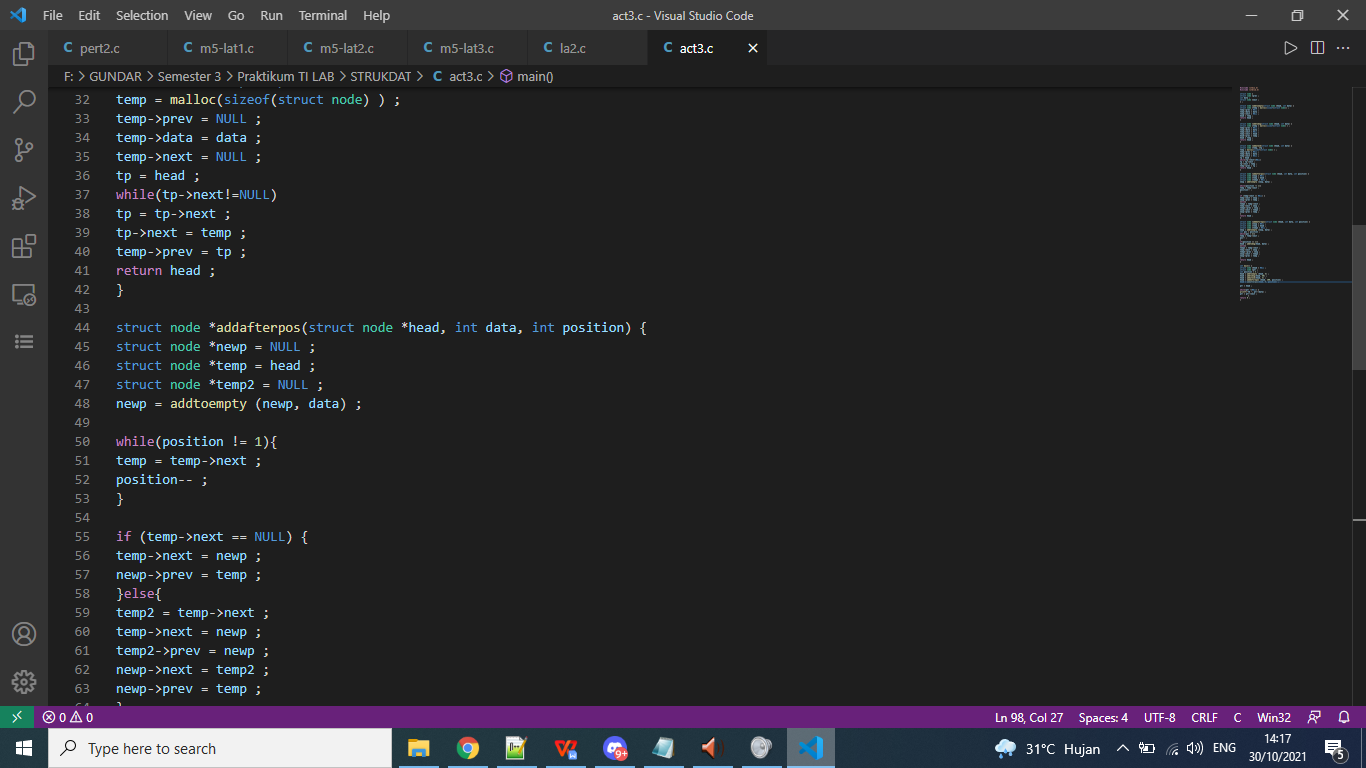
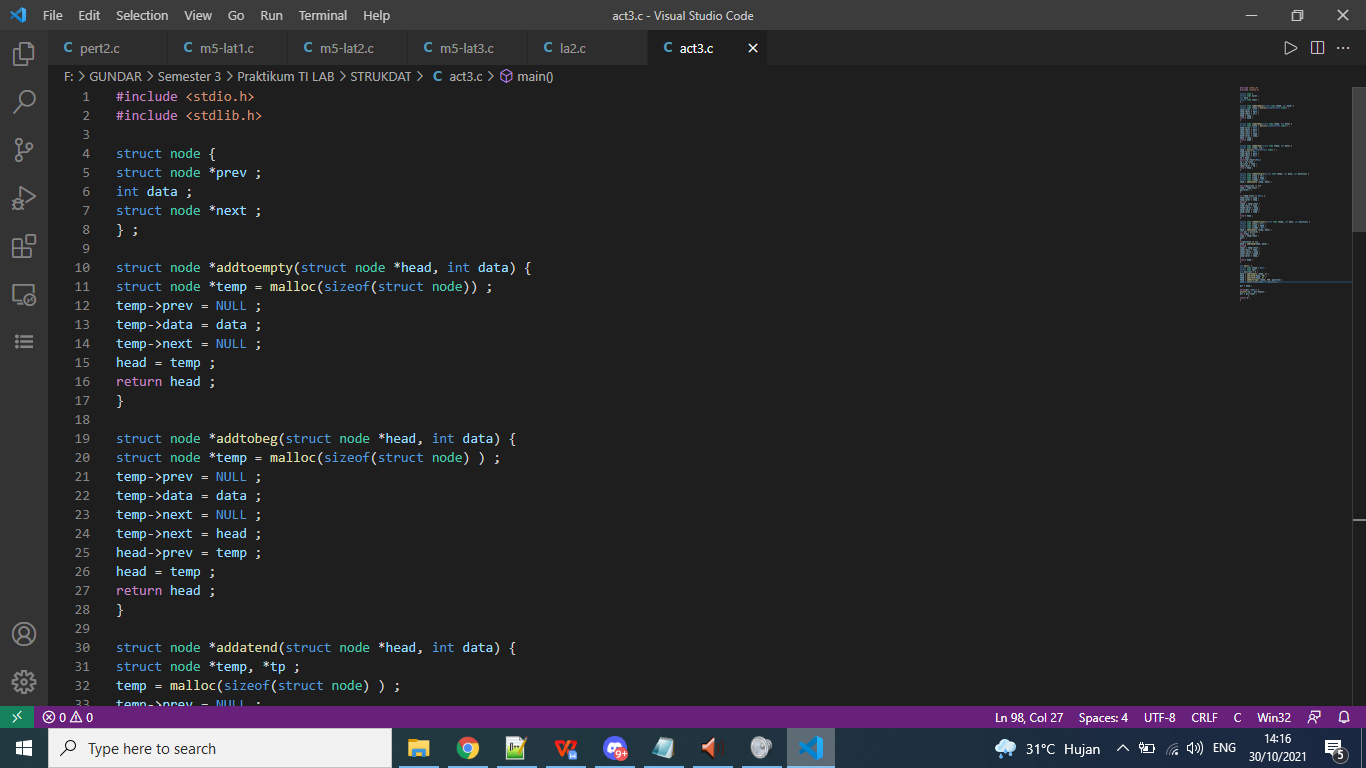
Jumlah Lembar : 10 Lembar



# LABORATORIUM TEKNIK INFORMATIKA UNIVERSITAS GUNADARMA

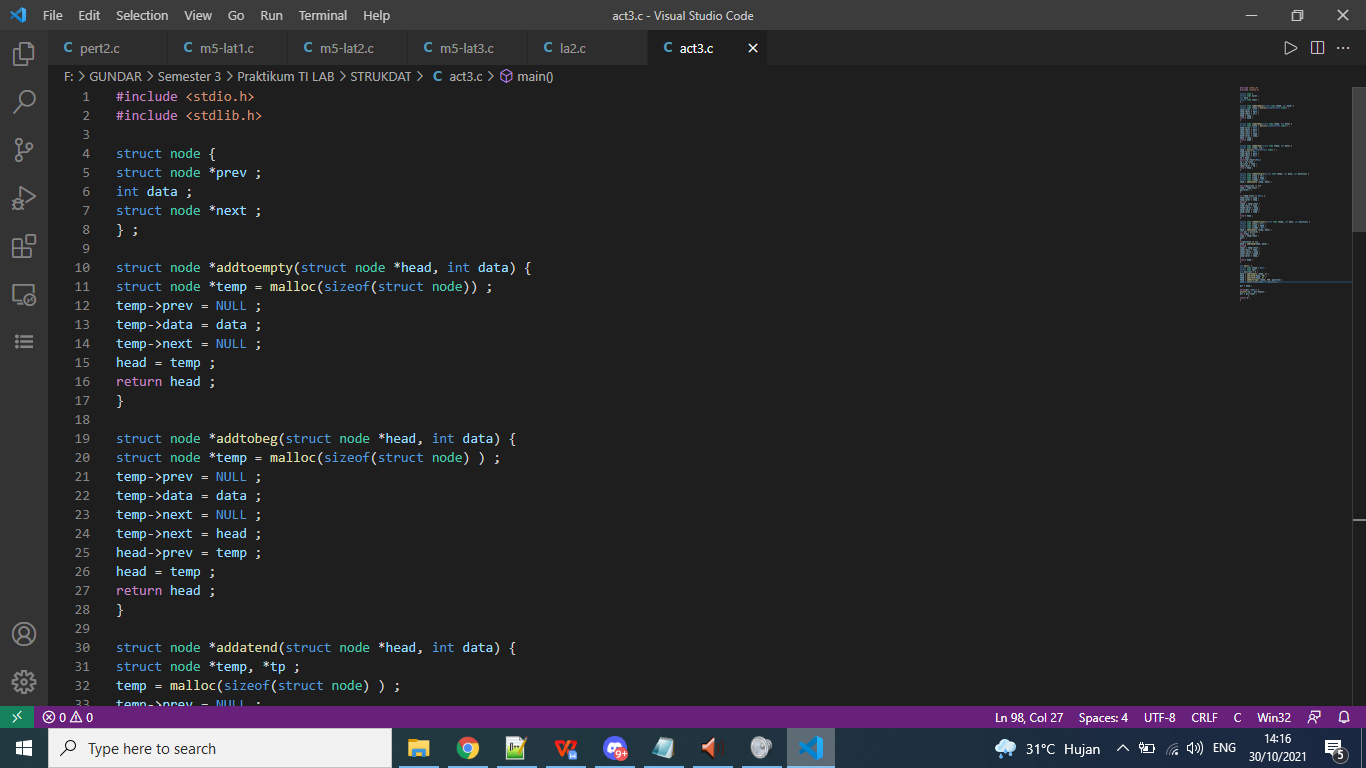
**2021**

**LISTING PROGRAM**

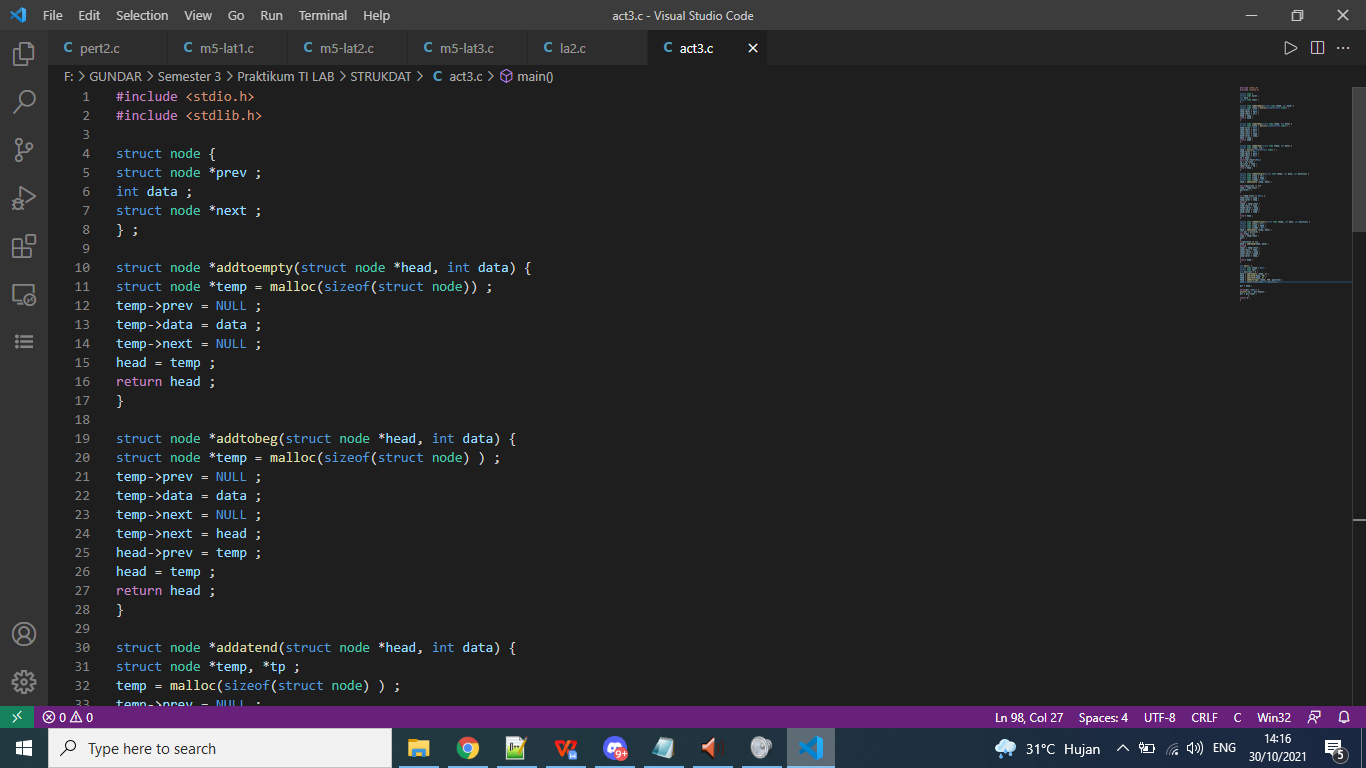


**LOGIKA PROGRAM**

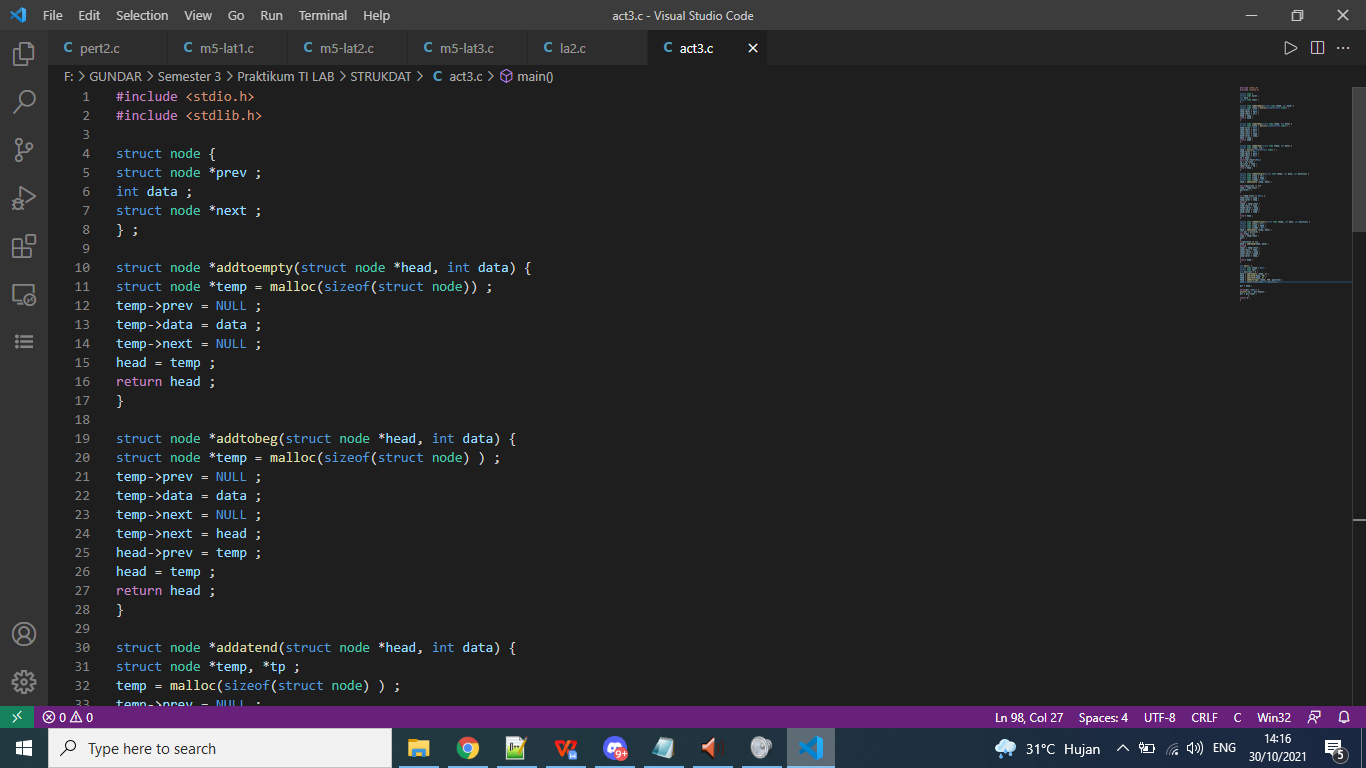
Pada Pertemuan Praktikum pertama ini yaitu Struktur Data, Saya mendapat materi Double Linked List. Untuk membuat program sederhana dari bahasa C. Sebelumnya kita akan membahas C singkat. Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie 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 Double Linked List dari NPM. Berikut Langkah Langkah dibawah ini untuk membuat Program Double Linked List dari NPM dari bahasa C yang bisa anda di lihat dibawah ini.



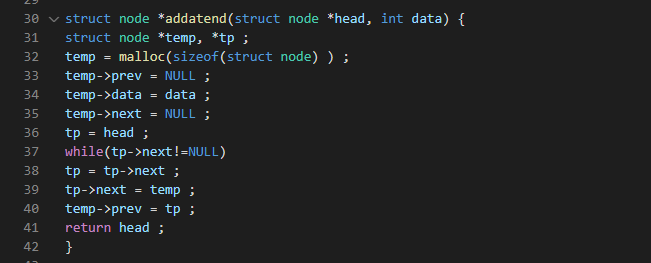
Langkah pertama kita harus memiliki text editor untuk menjalankan Bahasa C. yang saya gunakan disini ialah Aplikasi Visual Studio Code. Agar Bahasa C bisa di jalankan di Visual Studio Code harus meng install beberapa ekstensi terlebih dahulu dan saya menginstall juga MinGW. Yang bisa di download disini ( <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. setelah itu kita buat pointer untuk program ini dengan **struct node \*prev** dan **struct node \*next** .



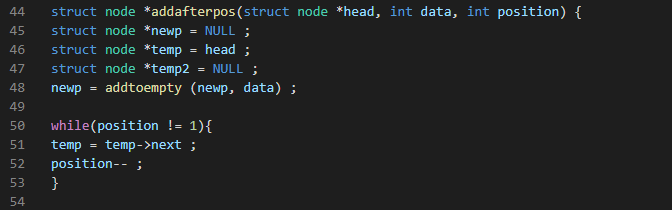
Langkah Selanjutnya adalah seperti gambar di atas pada baris ke lima yaitu berfungsi untuk **add to empty** pada program Lalu membuat dulu pointer **head** dengan tipe **integer** dengan program **(struct node \*head, int data) {** , dan buat juga pointer **temp** dengan **struct node \*temp = malloc(sizeof(struct node)) ;**. Lalu buat pointer **temp** menuju ke masing” pointer , contoh **temp->prev = NULL ;.** Dan membuat pointer **head** sama dengan **pointer temp .** Lalu setelah itu membuat **pointer head** Kembali ke fungsi utama dengan **return head ;**.



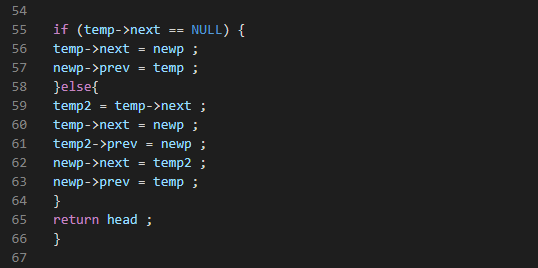
Langkah Selanjutnya adalah seperti gambar di atas yaitu adalah hampir sama dengan program sebelumnya. Tapi disini membuat **fungsi addtobeg** yang juga dengan **pointer head** dengan **tipe data integer** yaitu **struct node \*addtobeg(struct node \*head, int data) {.**



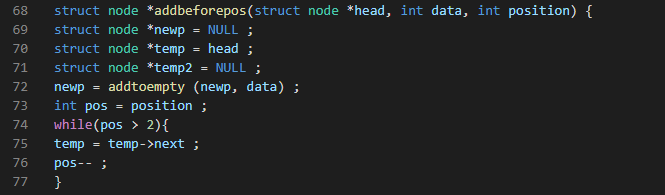
Langkah Selanjutnya adalah seperti gambar di atas yaitu **fungsi addatend** dengan **pointer head** dan juga tipe data integer. Lalu membuat pointer **temp** sama seperti program sebelumnya namun ada tambahan pointer untuk menyambungkan node-node setelah program ini dengan **struct node \*temp, \*tp ;.** Lalu membuat program untuk menyambungkan pointer temp ke pointer” lainnya dan membuat tp sama dengan pointer head. Lalu kita masuk ke perulangan. Selama **tp ke next tidak sama dengan null maka tp akan lanjut ke next** berikutnya dengan program **while(tp->next!=NULL) tp = tp->next ;**. Lalu kita tp ke next menjadi sama dengan pointer temp dengan program **tp- >next = temp ;.** Dan juga buat pointer temp ke prev menjadi sama dengan tp dengan program **temp->prev = tp ;**. Dan terakhir buat pointer head Kembali ke fungsi utama.



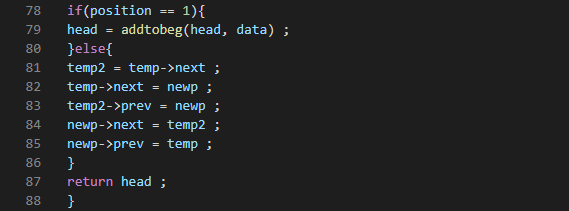
Langkah Selanjutnya adalah seperti gambar di atas yaitu membuat fungsi **addafterpos** dengan pointer head dan juga 2 buah variable data dengan integer yaitu data dan position. Lalu saya mendeklarasikan 3 pointer tambahan untuk setiap node yang ada yaitu **newp,temp dan temp2.** Lalu disini saya memakai pointer newp pada fungsi **addtoempty** dengan variable data dengan program **newp = addtoempty (newp, data) ;**. Lalu membuat perulangan selama variable position tidak sama dengan satu maka pointer temp akan sama dengan pointer temp ke next , dan juga variable position akan increment dengan program **while(position != 1){ temp = temp->next ; position--;** .



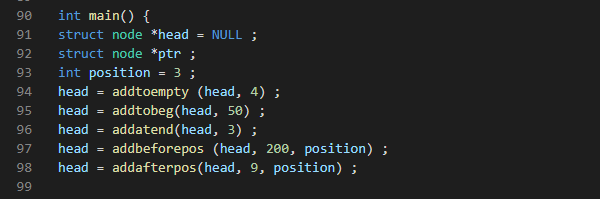
Langkah Selanjutnya adalah seperti gambar di atas yaitu kita masuk ke percabangan 2 kondisi , kondisi 1 adalah jika temp ke next sama dengan null maka **pointer temp ke next akan sama dengan pointer newp** , dan **newp ke prev akan menjadi sama dengan temp** dengan program **if (temp->next == NULL) { temp->next = newp ; newp->prev = temp ;**. Lalu masuk ke kondisi 2 dengan program **else{temp2 = temp->next ;temp->next = newp ; temp2->prev = newp ; newp->next = temp2 ; newp->prev = temp ; }**. Setelah itu Kembali ke fungsi utama dengan **return head**;



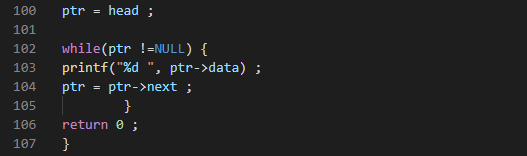
Langkah Selanjutnya adalah seperti gambar di atas yaitu untuk fungsi **addbeforepos** yang isi program nya hampir sama dengan kodingan fungsi **addafterpost** namun bedanya disini adalah ada pertukaran antara variable position menjadi variable pos dengan tipe data integer dengan kodingan **int pos = position ;.** Lalu membuat perulangan yaitu selama variable pos itu lebih dari 2 maka pointer temp akan sama dengan pointer temp ke next dan juga variable pos akan increment dengan program **while(pos > 2){ temp = temp->next ; pos-- ;}** .



Langkah Selanjutnya adalah seperti gambar di atas yaitu masih pada kodingan fungsi **addbeforepost.** Lalu membuat percabangan 2 kondisi dengan pertama yaitu jika **var position sama dengan 1** maka pointer head akan dimasukkan ke fungsi **addtobeg** dengan program **if(position == 1){ head = addtobeg(head, data) ;.** Lalu kondisi 2 itu untuk memasukkan pointer - pointer yang ada dengan program**}else{ temp2 = temp->next ; temp- >next = newp ; temp2->prev = newp ; newp->next = temp2 ; newp->prev = temp ; }** , lalu terakhir kembali ke fungsi utama dengan **return head;.**



Langkah Selanjutnya adalah seperti gambar di atas yaitu fungsi utama pada program ini. pertama kita membuat **pointer head = null** dengan kodingan **struct node \*head = NULL ;** dan juga membuat **pointer ptr** dengan program **struct node \*ptr ;** . Lalu saya deklarasikan nilai pointer head pada fungsi yang ada, pertama **int position = 3.** Lalu buat **fungsi empty pada head** itu menjadi **4** dengan program **head = addtoempty (head, 4) ;.** Lalu membuat fungsi **beg** pada head itu menjadi **50** yang dimana akan terletak sebelum empty dengan program **head = addtobeg(head, 50) ;.** Lalu membuat fungsi **end pada head** itu menjadi 3 yang dimana akan terletak paling terakhir dengan program **head = addatend(head, 3) ;** , buat fungsi **before pos pada head** itu menjadi **200** yang dimana akan terletak sebelum paling terakhir dengan program **head = addbeforepos (head, 200, position) ;.** Lalu yang terakhir fungsi **after pos pada head** itu menjadi **9** yang dimana akan terletak tepat sebelum paling terakhir dengan program **head = addafterpos(head, 9, position) ; .**



Selanjutnya adalah pada gambar terakhir yang di atas ini untuk mendeklarasikan **ptr itu sama dengan head.** Lalu masuk ke perulangan selama pointer ptr itu sama dengan null maka akan tercetak nilai **ptr menuju data dengan output berupa angka dan ptr sama dengan ptr sama dengan ptr menuju next** dengan program **while(ptr !=NULL) { printf("%d ", ptr->data) ; ptr = ptr->next ;.** Lalu untuk mengakhiri yaitu dengan program **return 0; }.**

**OUTPUT PROGRAM**

