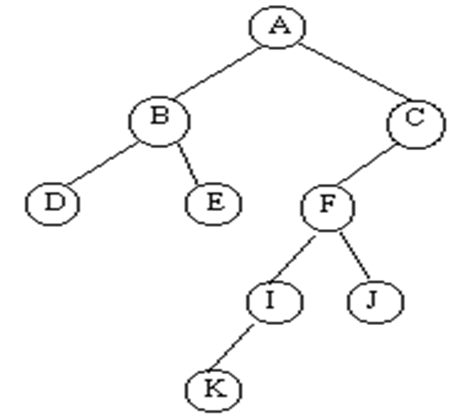
Praktikum

Tree

# NIM/Nama : 3311401066/Firchan Risanto, 3311401047/Dwi Teguh Prasetyo

|  |
| --- |
| Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:   * Menggambarkan konsep tree pada memory * Membuat struktur data tree dalam   bahasa C   * Melakukan operasi pada tree * Mengaplikasikan tree pada program |

Misalkan terdapat binary tree seperti contoh berikut.



Dengan memanfaatkan prosedur dan fungsi yang telah diberikan pada slide teori mengenai Tree, baris kode di main program yang membentuk tree tersebut adalah:

Simpul \*R;

inisialisasi(&R);

tambahSimpul(&R, 'A');

tambahSimpul(&R->kiri, 'B');

tambahSimpul(&R->kiri->kiri, 'D');

tambahSimpul(&R->kiri->kanan, 'E');

tambahSimpul(&R->kanan, 'C');

tambahSimpul(&R->kanan->kiri, 'F');

tambahSimpul(&R->kanan->kiri->kiri, 'I');

tambahSimpul(&R->kanan->kiri->kiri->kiri, 'K');

tambahSimpul(&R->kanan->kiri->kanan, 'J');

Tambahkanlah kode program untuk membaca tree tersebut secara preorder, inorder, dan postorder.

Salin kode C Anda di sini.

#include <stdio.h>

#include <stdlib.h>

struct Node{

char data;

struct Node \*kiri;

struct Node \*kanan;

};

typedef struct Node Simpul;

void inisialisasi(Simpul \*\*T) {

\*T = NULL;

}

Simpul \*simpulBaru(char item) {

Simpul \*n;

n = (Simpul\*) malloc(sizeof(Simpul));

if(n != NULL) {

n->data = item;

n->kiri = NULL;

n->kanan = NULL;

}

return n;

}

int isEmpty(Simpul \*T) {

return (T == NULL);

}

void tambahSimpul(Simpul \*\*p, char item) {

Simpul \*n;

n = simpulBaru(item);

\*p = n;

}

void preOrder(Simpul \*T) {

if(!isEmpty(T)) {

printf("%c ", T->data);

preOrder(T->kiri);

preOrder(T->kanan);

}

}

void inOrder(Simpul \*T) {

if(!isEmpty(T)) {

inOrder(T->kiri);

printf("%c ", T->data);

inOrder(T->kanan);

}

}

void postOrder(Simpul \*T) {

if(!isEmpty(T)) {

postOrder(T->kiri);

postOrder(T->kanan);

printf("%c ", T->data);

}

}

main(){

Simpul \*R;

inisialisasi(&R);

tambahSimpul(&R, 'A');

tambahSimpul(&R->kiri, 'B');

tambahSimpul(&R->kiri->kiri, 'D');

tambahSimpul(&R->kiri->kanan, 'E');

tambahSimpul(&R->kanan, 'C');

tambahSimpul(&R->kanan->kiri, 'F');

tambahSimpul(&R->kanan->kiri->kiri, 'I');

tambahSimpul(&R->kanan->kiri->kiri->kiri, 'K');

tambahSimpul(&R->kanan->kiri->kanan, 'J');

printf("Preorder\n");

preOrder(R);

printf("\nInorder\n");

inOrder(R);

printf("\nPostorder\n");

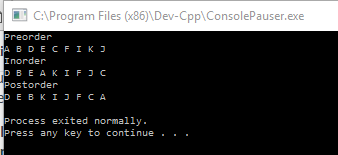
postOrder(R);

return 0;

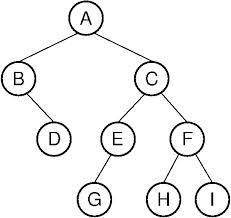
system("pause");

}

Jelaskan hasil running di sini.



Buatlah kode program untuk membuat tree seperti berikut dan menampilkan isinya secara preorder, inorder dan postorder.



Salin kode C Anda di sini.

#include <stdio.h>

#include <stdlib.h>

struct Node{

char data;

struct Node \*kiri;

struct Node \*kanan;

};

typedef struct Node Simpul;

void inisialisasi(Simpul \*\*T) {

\*T = NULL;

}

Simpul \*simpulBaru(char item) {

Simpul \*n;

n = (Simpul\*) malloc(sizeof(Simpul));

if(n != NULL) {

n->data = item;

n->kiri = NULL;

n->kanan = NULL;

}

return n;

}

int isEmpty(Simpul \*T) {

return (T == NULL);

}

void tambahSimpul(Simpul \*\*p, char item) {

Simpul \*n;

n = simpulBaru(item);

\*p = n;

}

void preOrder(Simpul \*T) {

if(!isEmpty(T)) {

printf("%c ", T->data);

preOrder(T->kiri);

preOrder(T->kanan);

}

}

void inOrder(Simpul \*T) {

if(!isEmpty(T)) {

inOrder(T->kiri);

printf("%c ", T->data);

inOrder(T->kanan);

}

}

void postOrder(Simpul \*T) {

if(!isEmpty(T)) {

postOrder(T->kiri);

postOrder(T->kanan);

printf("%c ", T->data);

}

}

main(){

Simpul \*R;

inisialisasi(&R);

tambahSimpul(&R, 'A');

tambahSimpul(&R->kiri, 'B');

tambahSimpul(&R->kiri->kanan, 'D');

tambahSimpul(&R->kanan, 'C');

tambahSimpul(&R->kanan->kiri, 'E');

tambahSimpul(&R->kanan->kiri->kiri, 'G');

tambahSimpul(&R->kanan->kanan, 'F');

tambahSimpul(&R->kanan->kanan->kiri, 'H');

tambahSimpul(&R->kanan->kanan->kanan, 'I');

printf("Preorder\n");

preOrder(R);

printf("\nInorder\n");

inOrder(R);

printf("\nPostorder\n");

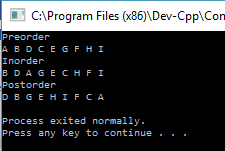
postOrder(R);

return 0;

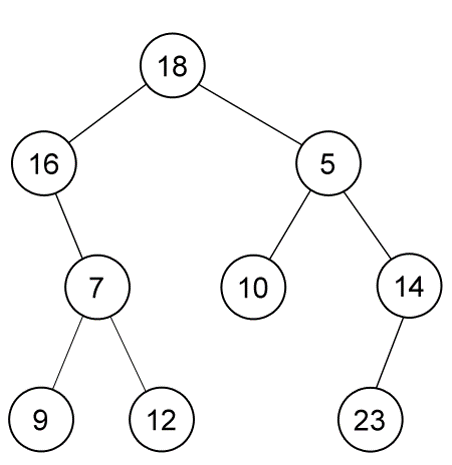
system("pause");

}

Jelaskan hasil running di sini.



Buatlah kode program untuk membuat tree seperti berikut dan menampilkan isinya secara preorder, inorder dan postorder. Modifikasi kode yang telah Anda miliki sehingga masing-masing simpul menyimpan data berupa satu bilangan bulat (integer).



Salin kode C Anda di sini. Jelaskan perubahan-perubahan yang Anda buat untuk mengakomodasi isi data berupa bilangan bulat.

#include <stdio.h>

#include <stdlib.h>

struct Node{

int data;

struct Node \*kiri;

struct Node \*kanan;

};

typedef struct Node Simpul;

void inisialisasi(Simpul \*\*T) {

\*T = NULL;

}

Simpul \*simpulBaru(int item) {

Simpul \*n;

n = (Simpul\*) malloc(sizeof(Simpul));

if(n != NULL) {

n->data = item;

n->kiri = NULL;

n->kanan = NULL;

}

return n;

}

int isEmpty(Simpul \*T) {

return (T == NULL);

}

void tambahSimpul(Simpul \*\*p, int item) {

Simpul \*n;

n = simpulBaru(item);

\*p = n;

}

void preOrder(Simpul \*T) {

if(!isEmpty(T)) {

printf("%d ", T->data);

preOrder(T->kiri);

preOrder(T->kanan);

}

}

void inOrder(Simpul \*T) {

if(!isEmpty(T)) {

inOrder(T->kiri);

printf("%d ", T->data);

inOrder(T->kanan);

}

}

void postOrder(Simpul \*T) {

if(!isEmpty(T)) {

postOrder(T->kiri);

postOrder(T->kanan);

printf("%d ", T->data);

}

}

main(){

Simpul \*R;

inisialisasi(&R);

tambahSimpul(&R, 18);

tambahSimpul(&R->kiri, 16);

tambahSimpul(&R->kiri->kanan, 7);

tambahSimpul(&R->kiri->kanan->kiri, 9);

tambahSimpul(&R->kiri->kanan->kanan, 12);

tambahSimpul(&R->kanan, 5);

tambahSimpul(&R->kanan->kiri, 10);

tambahSimpul(&R->kanan->kanan, 14);

tambahSimpul(&R->kanan->kanan->kiri, 23);

printf("Preorder\n");

preOrder(R);

printf("\nInorder\n");

inOrder(R);

printf("\nPostorder\n");

postOrder(R);

return 0;

system("pause");

}

Jelaskan hasil running di sini.

