Nama : Yanuar Nurul Hilal
NIM   : 222112418
Kelas : 2KS4

Penugasan Struktur Data Pertemuan 8

1. Modifikasi program pada Praktikum8A.c sehingga data yang disimpan pada Binary Search Tree, bukan lagi sebuah angka bertipe integer, melainkan data nama mahasiswa dengan tipe char[30] . Simpan hasil modifikasi Anda pada file Praktikum8B.c.

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

struct node
{
    char data[30];
    struct node *left;
    struct node *right;
};

struct node *newnode(char data[30])
{
    struct node *node = (struct node *)malloc(sizeof(struct node));
    strcpy(node->data, data);
    node->left = NULL;
    node->right = NULL;
    return node;
}

struct node *insert(struct node *root, char newdata[30])
{
    if (root == NULL)
    {
        root = newnode(newdata);
    }
    else
    {
        int is_left = 0;
        struct node *cursor = root;
        struct node *prev = NULL;
        while (cursor != NULL)
        {
            prev = cursor;
            if (strcmp(newdata, cursor->data) < 0)
            {
                is_left = 1;
                cursor = cursor->left;
            }
            else if (strcmp(newdata, cursor->data) > 0)
            {
                is_left = 0;
                cursor = cursor->right;
            }
        }
        if (is_left == 1)
            prev->left = newnode(newdata);
        else
            prev->right = newnode(newdata);
    }
    return root;
}

void searchnode(struct node *root, char data[30])
{
    int found = 1;
    struct node *cursor = root;
    while (strcmp(cursor->data, data) != 0)
    {
        if (cursor != NULL)
        {
            if (strcmp(data, cursor->data) > 0)
            {
                cursor = cursor->right;
            }
            else
            {
                cursor = cursor->left;
            }
            if (cursor == NULL)
            {
                printf("\nNode %s tidak ditemukan", data);
                found = 0;
                break;
            }
        }
    }
    if (found == 1)
        printf("\nNode %s ditemukan", data);
    return;
}

struct node *deletenode(struct node *root, char deleteddata[30])
{
    if (root == NULL)
        return 0;
    struct node *cursor;
    if (strcmp(deleteddata, root->data) > 0)
        root->right = deletenode(root->right, deleteddata);
    else if (strcmp(deleteddata, root->data) < 0)
        root->left = deletenode(root->left, deleteddata);
    else
    {
        // 1 CHILD
        if (root->left == NULL)
        {
            cursor = root->right;
            free(root);
            root = cursor;
        }
        else if (root->right == NULL)
        {
            cursor = root->left;
            free(root);
            root = cursor;
        }
        // 2 CHILDS NODE
        else
        {
            cursor = root->right;
            while (cursor->left != NULL)
            {
                cursor = cursor->left;
            }
            strcpy(root->data, cursor->data);
            root->right = deletenode(root->right, cursor->data);
        }
    }

    return root;
}

void displayPreorder(struct node *node)
{
    if (node == NULL)
        return;
    printf("%s ", node->data);    // root
    displayPreorder(node->left);  // subtree kiri
    displayPreorder(node->right); // subtree kanan
}

void displayInorder(struct node *node)
{
    if (node == NULL)
        return;
    displayInorder(node->left);  // subtree kiri
    printf("%s ", node->data);   // root
    displayInorder(node->right); // subtree kanan
}
void displayPostorder(struct node *node)
{
    if (node == NULL)
        return;
    displayPostorder(node->left);  // subtree kiri
    displayPostorder(node->right); // subtree kanan
    printf("%s ", node->data);     // root
}

int main()
{
    struct node *root = newnode("casillas");
    root = insert(root, "alves");
    root = insert(root, "ramos");
    root = insert(root, "nesta");
    root = insert(root, "maldini");
    root = insert(root, "zidane");
    root = insert(root, "iniesta");
    root = insert(root, "ronaldinho");
    root = insert(root, "cristiano");
    root = insert(root, "ronaldo");
    root = insert(root, "messi");
    printf("Tampilan Preorder : ");
    displayPreorder(root);
    printf("\nTampilan Inorder : ");
    displayInorder(root);
    printf("\nTampilan Postorder : ");
    displayPostorder(root);
    searchnode(root, "maguire");
    printf("\nDelete Node ramos");
    root = deletenode(root, "ramos");
    printf("\nInorder :");
    displayInorder(root);
    printf("\n");
    return 0;
}
```