|  |  |
| --- | --- |
| Nama | : La Ode Muhammad Gazali |
| NIM | : 222212696 |
| Kelas | : 2KS2 |

**MODUL 8 PRAKTIKUM STRUKTUR DATA**

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.

* Program hasil modifikasi

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct node

{

char data[30];

struct node \*left;

struct node \*right;

};

struct node \*newNode(const char\* data)

{

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

strncpy(node->data, data, sizeof(node->data));

node->left = NULL;

node->right = NULL;

return node;

}

struct node\* insert(struct node\* root, const char\* newData)

{

if (root == NULL) {

root = newNode(newData);

}

else {

int compare = strcmp(newData, root->data);

if (compare < 0) {

root->left = insert(root->left, newData);

}

else if (compare > 0) {

root->right = insert(root->right, newData);

}

}

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

}

void search\_node(struct node\* root, const char\* data)

{

struct node\* cursor = root;

while (cursor != NULL) {

int compare = strcmp(data, cursor->data);

if (compare == 0) {

printf("\nNode %s ditemukan", data);

return;

}

else if (compare < 0) {

cursor = cursor->left;

}

else {

cursor = cursor->right;

}

}

printf("\nNode %s tidak ditemukan", data);

}

struct node\* delete\_node(struct node\* root, const char\* deletedData)

{

if (root == NULL)

return root;

int compare = strcmp(deletedData, root->data);

if (compare < 0) {

root->left = delete\_node(root->left, deletedData);

}

else if (compare > 0) {

root->right = delete\_node(root->right, deletedData);

}

else {

if (root->left == NULL) {

struct node\* cursor = root->right;

free(root);

root = cursor;

}

else if (root->right == NULL) {

struct node\* cursor = root->left;

free(root);

root = cursor;

}

else {

struct node\* cursor = root->right;

while (cursor->left != NULL) {

cursor = cursor->left;

}

strncpy(root->data, cursor->data, sizeof(root->data));

root->right = delete\_node(root->right, cursor->data);

}

}

return root;

}

int main()

{

struct node\* root = newNode("Jordan");

root = insert(root, "Dwinanda");

root = insert(root, "Atikah");

root = insert(root, "Gazali");

root = insert(root, "Syawal");

printf("====Tampilan node awal=====");

printf("\nPreorder : "); displayPreorder(root);

printf("\nInorder : "); displayInorder(root);

printf("\nPostorder : "); displayPostorder(root);

printf("\n\n====Pencarian====");

search\_node(root, "Gazali");

search\_node(root, "Ilham");

root = delete\_node(root, "Dwinanda");

printf("\n\n===Setelah mengahapus Dwinanda===\n");

printf("Preorder : "); displayPreorder(root);

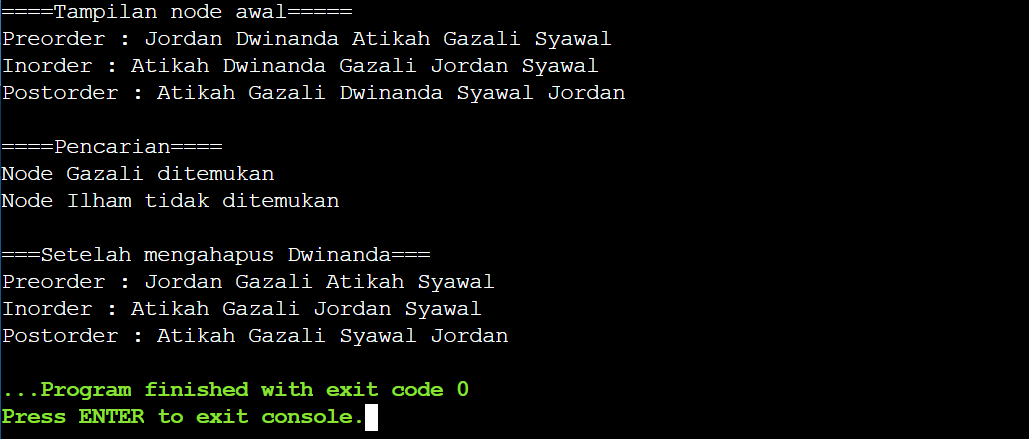
printf("\nInorder : "); displayInorder(root);

printf("\nPostorder : "); displayPostorder(root);

return 0;

}

* Output



* Penjelasan

Adapun beberapa modifikasi yang dilakukan agar binary search tree tersebut dapat menerima data bertipe char antara lain:

1. Menambahkan header #include <string.h>. Header tersebut berisi deklarasi fungsi-fungsi yang berkaitan dengan manipulasi string, serta konstanta-konstanta yang digunakan dalam pemrograman dengan string.
2. Menggunakan fungsi strcpy yang berfungsi untuk mengkopi sebagian dari satu string ke dalam string lain dengan batasan panjang tertentu.
3. Menggunakan fungsi strcmp untuk membandingkan dua string. Fungsi ini membandingkan dua string karakter demi karakter, hingga menemukan perbedaan atau mencapai akhir kedua string, sehingga dapat digunakan untuk mengurutkan data mahasiswa yang bertipe char pada binary search tree.
4. Penggunaan const pada parameter bertipe char yang bertujuan untuk menandai parameter tersebut sebagai "konstan" yang tidak akan diubah di dalam fungsi,