

Laporan Resmi

Algoritma Struktur Data

14.2 Tree



Nama: Gagas Amukti Nandaka
Kelas: 1 D4 Teknik Informatika B
NRP: 3120600032

PROGRAM STUDI TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2021

PERCOBAAN

1. Buatlah 2 buah variable bertipe stack, misalnya stOperator untuk menyimpan data operator dan stOperan untuk menyimpan data operan.
2. Masukkan ekspresi aritmatika dalam notasi infix, misalnya disimpan dalam string S. Hitung panjang string tsb, misalnya sebanyak N karakter
3. Dimulai dari $i = 1$ sampai N kerjakan langkah-langkah sebagai berikut:
 - a. $R = S[i]$
 - b. Test nilai R. Jika R adalah:
 - kurung buka : push ke dalam stack Operator
 - operand : push ke dalam stack Operan
 - operator : push ke dalam stack Operator
 - kurung tutup :
 - pop operator, bentuk nodenya & jadikan parent
 - pop operan, bentuk nodenya, taruh di kanan
 - pop operan, bentuk nodenya, taruh di kiri
 - push operator di stack Operan
 - c. Akhir notasi infix telah tercapa. Selama stack Operator dan stack Operan belum kosong, lakukan
 - pop operator, bentuk nodenya, & jadikan parent
 - pop operan, bentuk nodenya, taruh di kanan
 - pop operan, bentuk nodenya, taruh di kiri
4. Buatlah definisi preorder(), inorder() dan postorder() untuk mengimplementasikan algoritma masing-masing kunjungan preorder, inorder, dan postorder.
5. Lakukan pembacaan binary tree yang sudah terbentuk secara preorder, inorder dan postorder

```
MEMBENTUK BINARY TREE DARI SEBUAH EKSPRESI ARITMATIKA
DENGAN MEMANFAATKAN STRUKTUR DATA STACK

Masukan ekspresi dalam notasi infix : (9-5)*(7+4)

Hasil kunjungan secara PREORDER adalah : * - 9 5 + 7 4
Hasil kunjungan secara INORDER adalah : 9 - 5 * 7 + 4
Hasil kunjungan secara POSTORDER adalah : 9 5 - 7 4 + *

Process returned 0 (0x0)   execution time : 9.306 s
Press any key to continue.
```

Listing Program :

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>


typedef struct node tree;

struct node{

    tree*kiri;

    tree*kanan;

    char data;

};


typedef struct{

    tree*data[50];

    int count;

}stack;


void insert();

tree*pop(stack*);

int derajat(char);

int kosong(stack*);

int penuh(stack*);

void inisialisasi(stack*);

void preorder(tree*);

void inorder(tree*);

void boking(char);

void push(tree*,stack*);

void postorder(tree*);

tree*p,*temp;

tree*bantuan;

stack operat,operan;
```

```

int main()
{
    int pilihan, i;
    char notasi[50];
    inisialisasi(&operan);
    inisialisasi(&operat);

    puts("PRAKTIKUM TREE ARITMATIKA 14.2\n");
    printf("Masukkan notasi inputan : ");
    gets(notasi);

    for(i = 0; i < strlen(notasi); i++){
        fflush(stdin);
        boking(notasi[i]);
        bantuan = p;

        if(notasi[i] == '('){
            push(bantuan, &operat);
        }
        else if(notasi[i] == ')'){
            temp = pop(&operat);
            temp->kanan = pop(&operan);
            temp->kiri = pop(&operan);
            push(temp, &operan);
            pop(&operat);
        }
        else
            if(notasi[i] == '*' || notasi[i] == '^' || notasi[i] == '/' || notasi[i] == '+' || notasi[i] == '-' || notasi[i] == '-') {
                push(bantuan, &operat);
            }
    }
}

```

```

        else
if((notasi[i]>='A'&&notasi[i]<='Z')||(notasi[i]>='0'&&notasi[i]<='9')){
    push(bantuan, &operan);
}
else
    puts("Notasi inputan salah gan");
}
while((!kosong(&operan))&&(!kosong(&operat))){
    temp = pop(&operat);
    temp->kanan = pop(&operan);
    temp->kiri = pop(&operan);
}

do{
    puts("\nMENU TREE");
    puts("1. Preorder");
    puts("2. Postorder");
    puts("3. Inorder");
    puts("4. Exit");
    printf("masukkan pilihan anda : ");
    scanf("%d", &pilihan);
    fflush(stdin);

    if(pilihan == 1){
        puts("Hasil Preorder : ");
        preorder(temp);
        printf("\n");
    }
    else if(pilihan == 2){
        puts("Hasil Postorder : ");
        postorder(temp);
        printf("\n");
    }
}

```

```

    }
    else if(pilihan == 3){
        puts("Hasil Inorder : ");
        inorder(temp);
        printf("\n");
    }
    else if(pilihan == 4){
        puts("Arigatou gonzaimasu");
        exit(0);
    }
    else{
        puts("Masukkan menu salah");
    }
}while(1);
}

void boking(char input){
    p = (tree*)malloc(sizeof(tree));
    if(p == NULL){
        puts("Alokasi gagal, maaf");
        exit(0);
    }
    else{
        p->data = input;
        p->kiri = NULL;
        p->kanan = NULL;
    }
}

void push(tree*x, stack*s){
    if(penuh(s)){
        puts("Stack penuh, data tidak dapat disimpan");
    }else{

```

```

        s->data[s->count] = x;

        s->count++;

    }
}

tree*pop(stack*s){
    tree*temp;

    if(kosong(s)){
        puts("Saat ini Stack Kosong, tidak dapat mengambil data.");
    }else{
        s->count--;
        temp = s->data[s->count];
        return(temp);
    }
}

void inisialisasi(stack*s){
    s->count = 0;
}

int kosong(stack*s){
    if(s->count==0)
        return(1);
    else
        return(0);
}

int penuh(stack*s){
    if(s->count==50)
        return(1);
    else
        return(0);
}

int derajat(char masuk){
    if(masuk == '^')

```

```

        return 4;
    else if(masuk== '*' || masuk=='/')
        return 3;
    else if(masuk=='+' || masuk=='-')
        return 2;
    else if(masuk=='(')
        return 1;
    else
        return 0;
}

void preorder(tree*node) {
    if (node!=NULL) {
        printf("%c ",node->data);
        preorder(node->kiri);
        preorder(node->kanan);
    }
}

void inorder(tree *node){
    if (node!=NULL) {
        inorder(node -> kiri);
        printf("%c ",node -> data);
        inorder(node -> kanan);
    }
}

void postorder(tree *node){
    if (node!=NULL) {
        postorder(node -> kiri);
        postorder(node -> kanan);
        printf("%c ",node -> data);
    }
}

```


Capture Output :

```
"C:\Users\amukt\Dropbox\code\tree postfix\bin\Debug\tree postfix.exe"
PRAKTIKUM TREE ARITMATIKA 14.2

Masukkan notasi inputan : (9-5)*(7+4)

MENU TREE
1. Preorder
2. Postorder
3. Inorder
4. Exit
masukkan pilihan anda : 1
Hasil Preorder :
* - 9 5 + 7 4

MENU TREE
1. Preorder
2. Postorder
3. Inorder
4. Exit
masukkan pilihan anda : 2
Hasil Postorder :
9 5 - 7 4 + *

MENU TREE
1. Preorder
2. Postorder
3. Inorder
4. Exit
masukkan pilihan anda : 3
Hasil Inorder :
9 - 5 * 7 + 4

MENU TREE
1. Preorder
2. Postorder
3. Inorder
4. Exit
masukkan pilihan anda : 4
Arigatou gonzaimasu

Process returned 0 (0x0)   execution time : 32.689 s
Press any key to continue.
```