



Praktikum Struktur Data

Modul PSDA

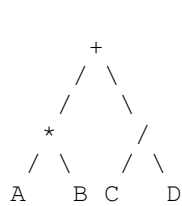
Tugas Aplikasi Queue

Dalam sesi praktikum ini, anda diminta membuat aplikasi dari ADT Stack dan atau ADT Queue yang sudah di pelajari pada sesi sebelumnya. Referensi yang dapat di pakai misalnya sebagai berikut :

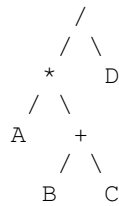
Konversi antara notasi

Infix	Postfix	Prefix
$((A*B) + (C/D))$	$((AB*) (CD/) +)$	$(+ (*AB) (/CD))$
$((A*(B+C)) / D)$	$((A(BC+) *) D/)$	$(/ (/ *A (+BC)) D)$
$(A*(B+(C/D)))$	$(A(B(CD/) +) *)$	$(*A (+B (/CD)))$

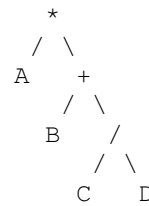
Trilk sederhana adalah dengan memahami konversi dari dank e dalam bentuk tree – tiap kurung merupakan membentuk suatu operasi dengan tiap operan mempunyai atasan operator sebagai sub-ekspresi



$((A*B) + (C/D))$



$((A*(B+C)) / D)$



$(A*(B+(C/D)))$

Silahkan pelajari kode di bawah ini :

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define MAX 20

char stack[MAX];
int top = -1;
char pop();
void push(char item);
int precd(char symbol) {
    switch(symbol) {
        case '+':
        case '-':
            return 2;
        case '*':
```



Praktikum Struktur Data

Modul PSDA

```
        case '/':
            return 4;
        case '^':
        case '$':
            return 6;
        case '(':
        case ')':
        case '#':
            return 1;
    }
}

int isoperator(char symbol) {
    switch(symbol) {
        case '+':
        case '-':
        case '*':
        case '/':
        case '^':
        case '$':
        case '(':
        case ')':
            return 1;
        default:
            return 0;
    }
}

void convertip(char infix[],char prefix[]) {
    int i,symbol,j=0;
    char test[MAX];
    infix=strrev(infix);
    stack[++top]='#';
    for(i=0;i<strlen(infix);i++) {
        symbol=infix[i];
        if(isoperator(symbol)==0) {
            prefix[j]=symbol;
            j++;
        }else {
            if(symbol==')') {
                push(symbol);
            }else if(symbol=='(') {
                while(stack[top]!='(') {
                    prefix[j]=pop();
                    j++;
                }
                pop();//pop out (.
            }
        }
    }
}
```



Praktikum Struktur Data

Modul PSDA

```
        }else {
            if(prcd(symbol)>prcd(stack[top])) {
                push(symbol);
            }else {
                while(prcd(symbol)<=prcd(stack[top]))
                {
                    prefix[j]=pop();
                    j++;
                }
                push(symbol);
            }
        }
    }
}

int main() {
    char infix[20],prefix[20];
    clrscr();
    printf("Masukan string infix:\n");
    gets(infix);
    convertip(infix,prefix);
    printf("String Perfix adalah:\n");
    puts(prefix);
    getch();
    return 0;
}

void push(char item) {
    top++;
    stack[top]=item;
}

char pop() {
    char a;
    a=stack[top];
    top--;
    return a;
}
```



Praktikum Struktur Data

Modul PSDA

Aplikasi yang di bangun adalah :

1. Memodifikasi Aplikasi Stack (evaluasi ekspresi) terdahulu dari **postfix** menjadi aplikasi **evaluasi ekspresi prefix** sebagai *input*. Contoh :

Input : $/*4+235$ atau $/(*4(+23))5$

Output: 4 karena $4*(2+3)/5$

2. Aplikasi yang kedua adalah membuat program memanfaatkan ADT Stack dan atau ADT Queue untuk memeriksa keseimbangan token kurung '(' dan ')' dalam suatu **ekspresi infix**. Untuk aplikasi ini silahkan eksplorasi algoritma yang anda pakai, tetapi untuk struktur data yang di gunakan **DI USAHAKAN** memakai ADT Stack dan atau ADT Queue yang sudah di pelajari. Contoh :

Input : $(((2+3)*4)/5)$

Output: $(((2+3)*4)/5)$ adalah balance

Input : $(((2+3*4)/5)$

Output: $(((2+3*4)/5)$ adalah tidak balance

Input : $((2+3)*4/5)$

Output: $((2+3)*4/5)$ adalah tidak balance