

**DEWI PUTRI AYUNINGSIH**

**A12.2019.06284**

**ALOGARITMA DAN PEMROGRAMAN / A12.6302**

**TUGAS STACK KONVERSI INFIX TO POSTFIX**

A=5, B=4, C=2, D=1.

1.  $(A+B) / (C-D) == AB+CD-/$

$(5+4)/(2-1)$

```
//header program.....
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <stdlib.h> // for exit().....
#include <ctype.h> // for isdigit(char ).....
#include <string.h>

#define SIZE 100

char stack[SIZE];
int top = -1;

//define push operation.....
void push(char item)
{
    if (top >= SIZE-1)
    {
        cout << "\nStack Penuh ! ";
    }
    else
    {
        top = top+1;
        stack[top] = item;
    }
}

//define pop operation.....
char pop()
{
    char item ;

    if (top < 0) {
        cout << "Stack Kosong ! ";
        getchar();
        exit(1);
    }
    else {
        item = stack[top];
        top = top-1;
    }
    return(item);
}
```

C:\BC5\LIB\INFIX YERE.exe

Program Konvesi Infix ke Postfix

Masukkan notasi Infix : (A+B)/(C-D)

Postfix yang terbentuk : AB+CD-/

2.  $A + B / C - D == ABC/+D-$

$5 + 4 / 2 - 1$

```
//operator yang diijinkan.....
int is_operator(char symbol)
{
    if (symbol == '^' || symbol == '*' || symbol == '/' ||
        symbol == '+' || symbol == '-')
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

//mengecek derajat operator.....
int precedence(char symbol)
{
    if (symbol == '^')
    {
        return (3);
    }
    else if (symbol == '*' || symbol == '/')
    {
        return (2);
    }
    else if (symbol == '+' || symbol == '-')
    {
        return (1);
    }
    else
    {
        return (0);
    }
}
}
```

```
C:\BC5\LIB\INFIX YERE.exe
Program Konvesi Infix ke Postfix
Masukkan notasi Infix : A+B/C-D
Postfix yang terbentuk : ABC/+D-
```

3.  $A * B + C / D == AB * CD /+$

$5 * 4 + 2 / 1$

```
//proses konversi.....
void InfixToPostfix(char infix_exp[], char postfix_exp[])
{
    int i, j;
    char item; //menampung karakter yg dibaca
    char x;

    push('(');
    strcpy(infix_exp, "");
    i = 0;
    j = 0;
    item = infix_exp[i];

    while (item != '\0')
    {
        //jika yg dibaca kurung buka
        if (item == '(')
        {
            push(item);
        }
        else
        {
            //jika yg dibaca operand
            if (isdigit(item) || isalpha(item))
            {
                postfix_exp[j] = item;
                j++;
            }
            else //jika yg dibaca operator
            {
                if (is_operator(item) == 1)
                {
                    x = pop();
                    while (is_operator(x) == 1 && precedence(x) >= precedence(item))
                    {
                        postfix_exp[j] = x;
                        j++;
                        x = pop();
                    }
                    push(x);
                }
                push(item);
            }
        }
        item = infix_exp[i++];
    }
}
```

```
C:\BC5\LIB\INFIX YERE.exe
Program Konvesi Infix ke Postfix
Masukkan notasi Infix : A*B+C/D
Postfix yang terbentuk : AB*CD/+
```

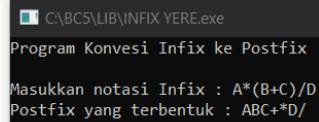
4.  $A * (B + C) / D == ABC + *D$

$5 * (4 + 2) / 1$

```
else
//jika yg dibaca kurung tutup
if(item == ')')
{
    x = pop();
    while(x != '(')
    {
        postfix_exp[j] = x;
        j++;
        x = pop();
    }
} else
//jika terdapat spasi
if(item == ' ')
{
    //spasi diabaikan
}
else{
cout << "\nEkspresi infix tidak valid ! \n";
getchar();
exit(1);}
i++;
item = infix_exp[i]; //ambil karakter berikutnya
} //akhir while
if(top>0)
{
    cout << "\nEkspresi infix tidak valid ! \n";
    getchar();
    exit(1);}
postfix_exp[j] = '\0';
//program utama.....
void main()
{
    char infix[SIZE], postfix[SIZE];
    cout << "Program Konvesi Infix ke Postfix\n";
    cout << "\nMasukkan notasi Infix : ";
    gets(infix);

    //konversikan infix ke postfix.....
    InfixToPostfix(infix, postfix);
    cout << "Postfix yang terbentuk : ";
    cout << postfix;

    getch();
}
```



C:\BC5\LIB\INFIX YERE.exe

Program Konvesi Infix ke Postfix

Masukkan notasi Infix : A\*(B+C)/D

Postfix yang terbentuk : ABC+\*D/