LAPORAN PRAKTIKUM

Algoritma dan Struktur Data

Week 2 : STACK

MUHAMMAD FARIS ISA

D4LJ – Teknik Informatika

3122640005

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2022

1. Latihan 3 & 4

Buatlah sebuah program yang melakukan pembalikan terhadap kalimat dengan menggunakan stack. Tentukan juga apakah kalimat yang diinputkan merupakan sebuah palindrom atau bukan.

- Source code :

```
#include<stdio.h>
#include<string.h>
#define MAX 50

typedef int Itemtype;
typedef struct{
    Itemtype item[MAX];
    int count;
} Stack;

void Inisialisasi(Stack *s){
    s->count = 0;
}

int Penuh(Stack *s){
    return s->count == MAX;
}

int Kosong(Stack *s){
    return s->count == 0;
}

void Push(Stack *s, Itemtype x){
    if (Penuh(s)) printf("Stack penuh!\n");
    else
        s->item[s->count]=x;
        s->count++;
}

Itemtype Pop(Stack *s){
    Itemtype temp = ' ';

    if(Kosong(s)) printf("Stack kosong!\n");
    else{
        s -> count--;
        temp = s -> item[s->count];
    }
}
```

```c
main(){
    Stack tump;

    char kalimat[MAX], kalimatBalik[MAX], hasil;
    int i, len, compare;
    Inisialisasi(&tump);

    printf("Masukkan kalimat : ");
    gets(&kalimat);

    len = strlen(kalimat);
    for(i = 0; i < len; i++){
        Push(&tump, kalimat[i]);
    }

    printf("Kebalikan dari kata %s ", kalimat);printf("adalah : ");
    for(i = 0; i < len; i++){
        hasil = Pop(&tump);
        printf("%c", hasil);
        kalimatBalik[i] = hasil;
    }

    kalimatBalik[len]='\0';
    printf("\n%s ", kalimatBalik);

    compare = strcmp(kalimat, kalimatBalik);
    if(compare == 0)
        printf("%s", "merupakan kata/kalimat polindrom" );
    else
        printf("%s", "bukan kata/kalimat polindrome");
}
```

- Hasil :

  a. Input dengan satu kata untuk menentukan apakah kalimat tersebut merupakan polindrom atau bukan.

```
a\Prak\Week2> & .\"Latihan3.exe"
Masukkan kalimat : sugus
Kebalikan dari kata sugus adalah : sugus
sugus merupakan kata/kalimat polindrom
```

```
a\Prak\Week2> & .\"Latihan3.exe"
Masukkan kalimat : kasur
Kebalikan dari kata kasur adalah : rusak
rusak bukan kata/kalimat polindrome
```

b. Input dengan dua kata untuk mengecek apakah kalimat tersebut merupakan polindrom atau bukan.

```
a\Prak\Week2> & .\"Latihan3.exe"
Masukkan kalimat : kasur rusak
Kebalikan dari kata kasur rusak adalah : kasur
rusak
kasur rusak merupakan kata/kalimat polindrom
```

```
a\Prak\Week2> & .\"Latihan3.exe"
Masukkan kalimat : malam rindang
Kebalikan dari kata malam rindang adalah : gnad
nir malam
gnadnir malam bukan kata/kalimat polindrome
```

- Analisa :
  1. `#include<string.h>` merupakan header yang digunakan untuk mendefinisikan satu buah tipe variabel, satu *macro* dan berbagai macam fungsi untuk memanipulasi array dari sebuah karakter. (C Library - string.h, 2022)
  2. `strlen(kalimat)` merupakan fungsi untuk menghitung panjang dari string. Tanda didalam kurung merupakan string yang ingin dihitung panjangnya. (C Library - string.h, 2022)
  3. `strcmp(kalimat, kalimatBalik);` merupakan fungsi yang digunakan utnuk membandingkan nilai string. Fungsi ini memiliki dua parameter (C Library - string.h, 2022). Pada contoh diatas variabel `kalimat` merupakan inputan yang diberikan oleh pengguna, sedangkan variabel `kalimatBalik` ini merupakan variabel yang disimpan pada saat sistem mengeluarkan nilai stack.
  4. Saat program dijalankan, pengguna diminta untuk melakukan input kalimat. Setelah itu sistem akan memproses kalimat dengan memasukkan setiap katanya kedalam stack. Selanjutnya untuk membalik kalimat, setiap yang ada di stack hanya perlu di pop dan di print.

     Untuk mencocokkan kalimat yang diinput dengan kalimat yang dibalik itu sama, pada saat melakukan proses pop nilai yang di pop harus disimpan ke dalam suatu variabel. Nilai tersebut disimpan disaat melakukan prulangan pada line `kalimatBalik[i] = hasil;`

     Setelah disimpan kalimat yang sudah dibalik, pada pemrograman c perlu adanya penanda bahwa suatu kata telah berhenti. Itu ditandai dengan adanya line `kalimatBalik[len]='\0';` Selanjutnya kalimatBalik ini nanti disamakan dengan kalimat yang sebelumnya diinput menggunakan fungsi `strcmp(kalimat, kalimatBalik)`. Apabila nilai yang dibandingkan sama, maka fungsi ini nanti akan mengembalikan nilai 0.

2. Latihan 6

Membuat implementasi dari input infix, lalu menampilkan operasi postfix dan juga menampilkan hasil dari perhitungan postfix

- Source code :

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#define MAX 100

typedef int Itemtype;
typedef struct{
    Itemtype item[MAX];
    int count;
} Stack;
Stack postfix;

void Inisialisasi(Stack *s)
{
    s->count=0;
}
int Penuh(Stack *s)
{
    return s->count==MAX;
}
int Kosong(Stack *s)
{
    return s->count==0;
}
void Push(Stack *s, Itemtype x)
{
    if(Penuh(s)) printf("Stack penuh!\n");
    else{
        s->item[s->count]=x;
        s->count++;
    }
}
Itemtype Pop(Stack *s)
{
    Itemtype temp=0;
    if(Kosong(s)) printf("Stack kosong!\n");
    else{
        s->count--;
        temp=s->item[s->count];
    }
        return temp;
}
```

```
int drjt(char x){ //menentukan derajat operator
    if(x == '(')
        return 0;
    else if((x == '+') || (x == '-'))
        return 1;
    else if((x == '*') || (x == '/'))
        return 2;
    else if(x == '^')
        return 3;
    else
        return -1; //invalid operator
}

char konversi_cetak(char temp[]) {
    Stack tump;
    int i, valid = 1, len = strlen(temp);
    char kar, smtr;
    Inisialisasi(&tump);
    Inisialisasi(&postfix);
    for ( i = 0 ; i < len ; i++ ) {
        kar = temp[i];
        switch(kar){
            case '(' :  Push(&tump, kar);
                        break;
            case '0': case '1': case '2': case '3': case '4': case '5'
            : case '6': case '7': case '8': case'9':
                        printf("%c", kar);
                        Push(&postfix, kar);
                        break;
            case '+': case '-': case'*': case '/': case '^' :
                        if((Kosong(&tump)) || ((drjt(kar) >
drjt(tump.item[tump.count-1])))
                            Push(&tump, kar);
                        else {
                            do {
                                smtr = Pop(&tump);
                                printf("%c", smtr);
                                Push(&postfix, smtr);
                            } while (drjt(kar) <
drjt(tump.item[tump.count-1]));
                                Push(&tump, kar);
                            }
                        break;
```

```
            case ')' :  while(tump.item[tump.count-1] != '(')
                            {
                                smtr = Pop(&tump);
                                printf("%c", smtr);
                                Push(&postfix, smtr);
                            }
                        smtr = Pop(&tump);
                        break;
            default :   valid = 0;
                        puts("INVALID STSTEMEN");
                        break;
        } //end of switch-case
    } //end of looping for

    for ( i = 0; i < len; i++){
        if((valid != 0) && (!Kosong(&tump))) {
                smtr = Pop(&tump);
                printf("%c", smtr);
                Push(&postfix, smtr);
        }
    }
}

main(){
    Stack tump;
    char kar, infix[MAX], postfix2[MAX];
    int OpLeft, OpRight, hasil, i, len;
    Inisialisasi(&tump);

    printf("Masukkan Notasi Infix : ");scanf("%s", &infix);
    printf("Hasil merubah notasi infix ke postfix = ");
    konversi_cetak(infix);

    len = postfix.count;
    for ( i = 0; i < len ; i++){
        kar = Pop(&postfix);
        postfix2[len-i-1] = kar;
    }

    postfix2[len]='\0';
```

```
    for( i = 0; i < strlen(postfix2); i++ ){
        kar = postfix2[i];
        if(kar >= '0' && kar <= '9')
            Push(&tump, kar-48);
        else if (kar == '+' || kar == '-' || kar == '*' || kar == '/' ||
kar == '^'){
            OpRight=Pop(&tump);
            OpLeft=Pop(&tump);
            switch (kar){
                case '+':   hasil = OpLeft + OpRight;
                            break;
                case '-':   hasil = OpLeft - OpRight;
                            break;
                case '*':   hasil = OpLeft * OpRight;
                            break;
                case '/':   hasil = OpLeft / OpRight;
                            break;
                case '^':   hasil = (int) pow(OpLeft, OpRight);
                            break;
            }
            Push(&tump, hasil);
        }
    }
    if(!Kosong(&tump)) printf("\nHasil operasi dari postfix
adalah %d\n", Pop(&tump));
}
```

- Hasil :

```
a\Prak\Week2> & .\"Latihan6.exe"
Masukkan Notasi Infix : (2+5)*(9-8)
Hasil merubah notasi infix ke postfix = 25+98-*
Hasil operasi dari postfix adalah 7
```

```
k\Week2> & .\"Latihan6.exe"
Masukkan Notasi Infix : (6-4)/2*2^3
Hasil merubah notasi infix ke postfix = 64-2/23^*
Hasil operasi dari postfix adalah 8
```

- Analisa :

1. Saat program dijalankan, program akan meminta pengguna untuk memasukkan Notasi Infix. Setelah itu notasi infix ini nanti akan di proses melewati rule seperti yang ditetapkan. Setelah menghasilkan notasi postfix, selanjutnya memproses notasi postfix yang sudah berhasil dibuat sebelumnya kedalam proses perhitungan. Pada for loop yang pertama, digunakan untuk menyimpan nilai-nilai yang

sebelumnya sudah disimpan ke dalam stack, dipindahkan ke dalam variable char string. Setelah itu `postfix2[len]='\0';` digunakan untuk menandakan bahwa string postfix2 ini berhenti di index `len`. For loop yang kedua digunakan untuk memproses postfix yang sebelumnya sudah disimpan ke dalam char string. Setelah itu akan langsung di print hasil dari perhitungan seperti contoh diatas.

KESIMPULAN

1. Stack merupakan sebuah kumpulan dari beberapa elemen data yang dapat diinput menggunakan push operation, dapat dihapus menggunakan pop operation berdasarkan sistem LIFO (Last In First Out) atau yang terakhir masuk merupakan yang pertama kali keluar.
2. Dengan menerapkan konsep LIFO, setiap perubahan baik menambahkan atau pengurangan terjadi di TOS (Top of Stack)
3. Pengaplikasian dari Stack di dalam C Programming dapat dengan merubah notasi infix menjadi notasi postfix.

DAFTAR PUSTAKA

C Library - string.h. (2022, September 3). Diambil kembali dari tutorialspoint: https://www.tutorialspoint.com/c_standard_library/string_h.htm