

TUGAS PRAKTIKUM ALGORITMA & STRUKTUR DATA

Jilid 9



Oleh :

Nama : Rosi Arif Mulyadi

NRP : 3121522021

Prodi : D3 Teknik Informatika PENS PSDKU Sumenep

Kelas : 1 ITA D3 Sumenep

Dosen :

LUSIANA AGUSTIEN M.Kom

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

MODUL 5

STACK / TUMPUKAN

C. Tugas Praktikum

1. Lakukan Modifikasi pada implementasi stack dengan Array. Dengan inputan yang di inputkan dari keyboard dalam bentuk satu kalimat utuh "Baca Buku", output yang dihasilkan berupa kalimat terbalik dari kalimat yang di inputkan "ukuB acaB"

Jawab :

Listing Program :

```
#include <stdio.h>
#include<string.h>
#include <conio.h>
#include<string.h>
#define MAKS 10
```

```
struct Tumpukan
{
    char s[10];
    int sp;
};typedef struct Tumpukan tmpkan;
tmpkan st;
```

```
int main()
{
    int pilihan;
    char item[10];
    do
    {
        printf("PERCOBAAN STACK \n\n");
        printf("MENU");
        printf("\n 1. PUSH");
        printf("\n 2. POP");
        printf("\n 3. TAMPIL");
        printf("\n 4. KELUAR");
```

```

printf("\n Inputkan Pilihan Anda : ");
scanf("%d", &pilihan);
fflush(stdin);

switch(pilihan)
{
    case 1:
        printf("\n Masukkan sebuah element ke atas tumpukan : ");
        scanf("%s", &item);
        //gets(item);
        push(item);
        Tampil();
        break;

    case 2:
        pop();
        Tampil();
        break;
    case 3:
        Tampil();
        break;
}
}while(pilihan != 4);
return 0;
getch();
}

int kosong()
{
    if(st.sp == -1)
    {
        printf("Tumpukan kosong\n");
        return 1;
    }
    else
        return 0;
}

int penuh()
{

```

```

if(st.sp == MAKS-1)
{
    printf("Tumpukan penuh");
    return 1;
}
else
    return 0;
}
int push(char nil[])
{
    int jawab;
    jawab=penuh();

    if(jawab==0)
    {
        st.sp++;
        strcpy(st.s[st.sp],nil);
    }else
    {
        printf("Tumpukan penuh");
    }
    return 0;
}
int pop()
{
    int jawab;
    jawab=kosong();

    if(jawab==0)
    {
        printf("\n Element teratas pada tumpukan adalah \t%s", st.s[st.sp]);
        st.sp--;
    }else
    {
        printf("\n Tumpukan kosong.tidak ada item yang dapat di hapus.");
    }
    return 0;
}

```

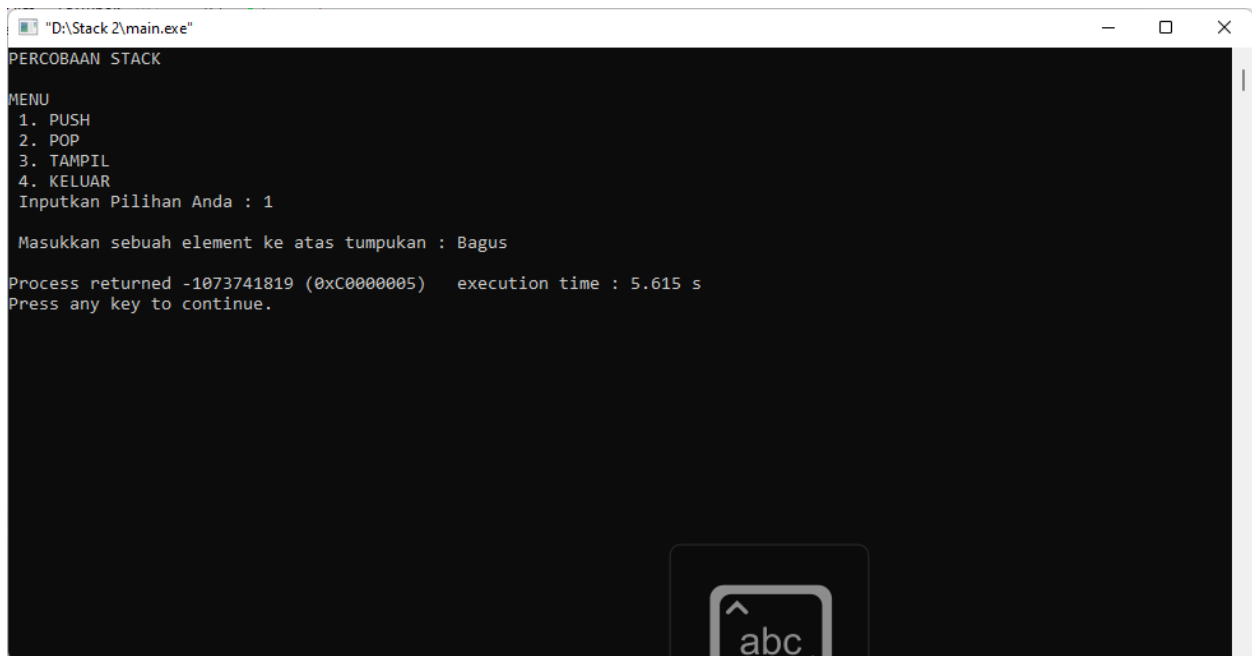
```

int Tampil()
{
    int i;
    printf("\n");

    if(kosong()==0)
    {
        printf("\n * Element-element pada tumpukan**");
        for(i=st.sp; i>=0; i--)
            printf("\n%s\n", st.s[1]);
    }else
    {
        printf("\n Tumpukan kosong.");
    }
    return 0;
}

```

Output :



```

D:\Stack 2\main.exe
PERCOBAAN STACK
MENU
1. PUSH
2. POP
3. TAMPIL
4. KELUAR
Inputkan Pilihan Anda : 1

Masukkan sebuah element ke atas tumpukan : Bagus

Process returned -1073741819 (0xC0000005)   execution time : 5.615 s
Press any key to continue.

```

2. Diketahui data data sebagai berikut: Manga, jambu, apel, dan Duku Buatlah algoritma dan program untuk memasukkan data-data diatas sehingga akan muncul tampilan sebagai berikut: Apel, duku, jambu, dan manga. Manfaatkan operasi push dan pop, tampilkan output secara ascending dan descending

Jawab :

```

#include<stdio.h>
#define BTS 15

//typedef int itemtype;

typedef struct
{
    char item['0'][BTS];
    int Count;
}stact;

void inisialisasi(stact *s){
    s->Count=0;
}

int empty(stact *s){
    return(s->Count == 0);
}

int full(stact *s){
    return(s->Count == BTS);
}

void push(char x[], stact *s){
    if (s->Count==BTS)
    {
        printf("Stact penuh! Data tidak bisa di masukkan!");
    }else{
        strcpy(s->item[s->Count],x);

        ++(s->Count);
    }
}

int pop(stact *s){
    if (s->Count==0)
    {

```

```

        printf("Stact masih kosong!");
    }else{
        --(s->Count);
        return(s->item[s->Count]);
    }
}

```

```

int muncul(stact *s){
    int i;
    printf("\n");
    printf("\n**Tampilan acak**\n");
    for(s->Count; s->Count>0; s->Count--){
        if(s->Count==3){
            return("%s",s->item[4]);
        }

        if(s->Count==4){
            return("%s",s->item[3]);
        }

        if(s->Count==2){
            return("%s",s->item[2]);
        }

        if(s->Count==1){
            return("%s",s->item[1]);
        }

        printf("-->");
        //printf("\n%s",st.s[i]);
    }
    printf(" | |");

    return 0;
}

```

```

int main(int argc, char const *argv[])
{
    int i;
    char input[BTS];
    stact tumpukan;
    inisialisasi(&tumpukan);
    for (i = 0; i < 3; ++i)
    {
        printf("Masukkan isi stact ke-%d :",i+1);
        scanf("%s",&input);
        push(input,&tumpukan);
    }

    for (i = 3; i > 0; --i)
    {
        printf("Isi stact ke-%d :",i);
        printf("%s\n",pop(&tumpukan) );
    }

    for(i=3; i>0; --i){
printf("%s\n",muncul(&tumpukan));
    }
    return 0;
}

```

Output :


```
"D:\Stack 4\main.exe"
Masukkan isi stact ke-1 :Duku
Masukkan isi stact ke-2 :Jeruk
Masukkan isi stact ke-3 :Mangga
Isi stact ke-3 :Mangga
Isi stact ke-2 :Jeruk
Isi stact ke-1 :Duku

**Tampilan acak**
||(null)

**Tampilan acak**
||(null)

**Tampilan acak**
||(null)

Process returned 0 (0x0)   execution time : 17.513 s
Press any key to continue.
_
```