

## Stack



Sisilia Thya Safitri, ST., MT

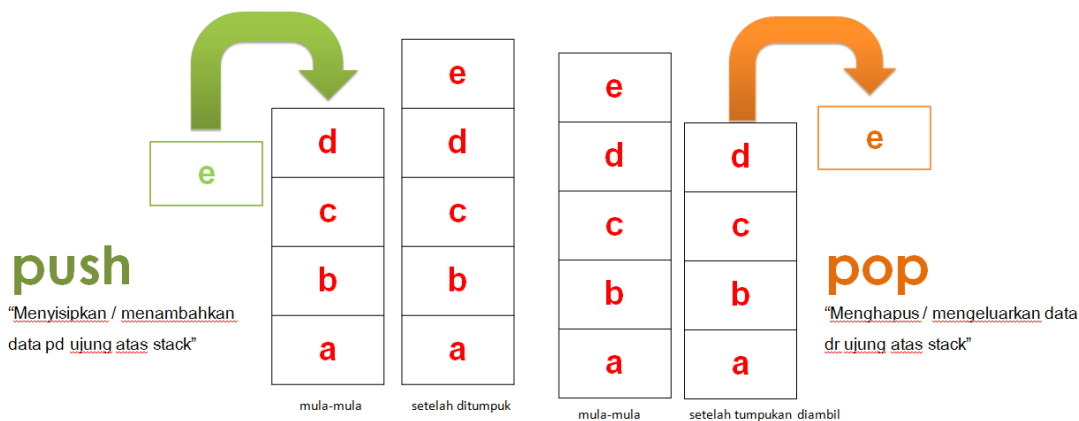
Jl. DI Panjaitan 128 Purwokerto

\* Untuk kalangan sendiri

**Praktikum 6**  
**Materi : Stack / Tumpukan**  
**Waktu : 100 menit**

**Dasar Teori**

“A stack is an ordered collection of items into which new items may be inserted and from which items may be deleted at one end, called top of the stack” \* Yedidyah L, Moshe J. A., and Aaron M. Tenenbaum; Data Structures Using C and C++. Secara sederhana, tumpukan bisa diartikan sebagai suatu kumpulan data yang seolah-olah ada data yang diletakan diatas data yang lain. Satu hal yang perlu kita ingat adalah bahwa kita bisa menambah (menyisipkan) data, dan mengambil (menghapus) data lewat ujung yang sama, yang disebut sebagai ujung atas tumpukan (*top of stack*). **LIFO ( Last In First Out )** adalah sifat dari stack data yang disimpan terakhir akan diambil lebih dahulu, data yang disimpan pertama kali akan diambil paling akhir”



Operasi Stack, Push and Pop

**Guided :**

1. Buatlah program dengan nama **GD1\_KELAS\_NIM**. Program ini merupakan program menu yang digunakan untuk melakukan proses PUSH dan POP pada sebuah stack

```
#include <iostream>
#include <conio.h>
#define max 10
#include <cstdlib>

using namespace std;

//inisialisasi Stack
struct Tumpukan{
    int atas;
    char data[max];
}T;

//inisialisasi T awal
void awal(){
    T.atas=-1;
}

//pengecekan Tumpukan apabila kosong
int kosong(){
    if(T.atas== -1)
        return 1;
    else
        return 0;
}

//pengecekan Tumpukkan apabila penuh
int penuh(){
    if(T.atas==max-1)
        return 1;
    else
        return 0;
}
```

```

//proses input data
void input(char data) {
    if(kosong()==1)
        {T.atas++;
          T.data[T.atas]=data;

          cout<<"Data "<<T.data[T.atas]<<" masuk ke stack";}
    else if(penuh()==0)
        {T.atas++;
          T.data[T.atas]=data;
          cout<<"Data "<<T.data[T.atas]<<" masuk ke stack";}

    else
        cout<<"Tumpukan penuh";
}

//proses hapus
void hapus() {
    if(kosong()==0) {
        cout<<"Data teratas sudah terambil";
        T.atas--;
    }
    else
        cout<<"Data kosong";
}

```

```

//Tampil data
void tampil() {
    cout<<"Isi Tumpukkan"<<endl;
    if(kosong()==0)

        {for(int i=T.atas;i>=0;i--)
            {
                cout<<T.data[i]<<endl;
            }
        }
    else
        cout<<"Tumpukan kosong";
}

//menghapus tumpukkan
void bersih() {
    T.atas=-1;

    cout<<"Tumpukan kosong!";
}

int main() {
    int pil;
    char data;
    awal();
    do
{

```

```

int main(){
    int pil;
    char data;
    awal();
    do
    {

        //program menu Stack
        system("CLS");
        cout<<"Menu STACK"<<endl;
        cout<<"1. push\n2. pop\n3. Tampil\n4. Bersihkan\n5. Keluar\nMasukkan pilihan :";
        cin>>pil;
        switch(pil)
        {
            case 1:cout<<"Masukkan data = ";cin>>data;
                    input(data);
                    break;
            case 2:hapus();
                    break;
            case 3:tampil();
                    break;
            case 4:bersih();
                    break;
            case 5: cout<<"Terimakasih, tekan enter untuk keluar";
                    }
        getch();
    }
    while(pil!=5);}

```

2. Buatlah program dengan nama **GD2\_KELAS\_NIM**. Program ini untuk melakukan pembalikan terhadap kalimat dengan menggunakan stack.

```

#include <iostream>
#include <conio.h>
#include <string.h>

using namespace std;

int main()
{
    char kata[35]="ST3 TELKOM PURWOKERTO";
    int inputkata;
    int i;

    cout<<"===STACK MEMBALIKAN KATA===\n";

    cout<<"Kata yang dimasukkan adalah: "<<kata<<endl;
    inputkata=strlen(kata); //strlen=menghitung jumlah karakter kata
    cout<<"Kata dibalik menjadi : ";
    for(i=inputkata;i>=1;i--)
    {
        char stack= kata[i-1];
        cout<<stack;
    }
    getch();
}

```

## Unguided

1. Buatlah program dengan nama **UGD\_KELAS\_NIM**. Program tersebut digunakan untuk menentukan apakah sebuah kalimat yang diinputkan dalam program (dengan menggunakan stack) adalah sebuah palindrom atau bukan. Palindrom adalah kalimat yang jika dibaca dari depan dan dari belakang, maka bunyinya sama.

Contoh:

Kalimat : sugus

Kalimat tersebut adalah palindrom

Kalimat : sisil

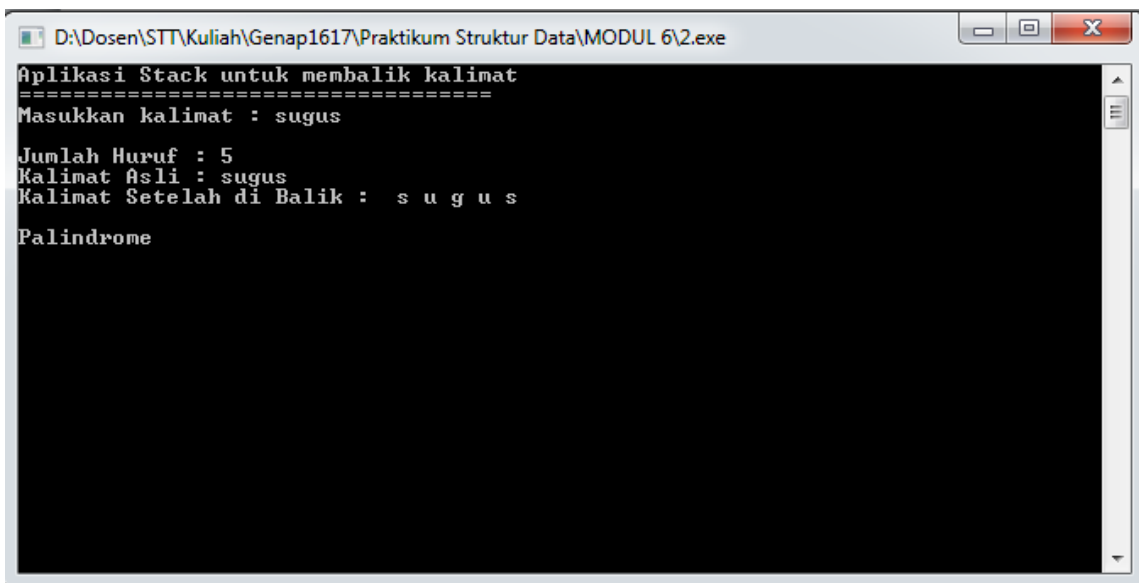
Kalimat tersebut bukan palindrom

Algoritma :

1. Mulai
2. Masukkan kata
3. Hitung jumlah pada inputan user
4. Masukkan ke dalam stack (push)
5. Bandingkan elemen 1 dalam stack dengan elemen terakhir (pop)
6. Perbandingan dilakukan berulang sebanyak jumlah huruf
7. Jika huruf yang dibandingkan semuanya sama, maka kata tersebut adalah palindrome
8. Selesai

Buatlah program dengan stack untuk mengubah notasi matematika infix menjadi postfix !

Output pada program adalah sebagai berikut:



```
D:\Dosen\STT\Kuliah\Genap1617\Praktikum Struktur Data\MODUL 6\2.exe
Aplikasi Stack untuk membalik kalimat
=====
Masukkan kalimat : sugus

Jumlah Huruf : 5
Kalimat Asli : sugus
Kalimat Setelah di Balik : s u g u s
Palindrome
```

