
**JURNAL PRAKTIKUM
(LAB. ACTIVITY)
STRUKTUR DATA
SI025**

**Materi 10 :
STACK
(TUMPUKAN)**

Dosen:

**Agung Nugroho, M.Kom
Prof. Dr. Ema Utami, S.Si, M.Kom
Ikmah, M.Kom
Ninik Tri H, M.Kom
Lilis Dwi Farida, M.Kom
Windha Mega PD, M.Kom**

**S1 - SISTEM INFORMASI
UNIVERSITAS AMIKOM YOGYAKARTA
2018**

Stack(Tumpukan)

Pendahuluan

A. Tujuan

Setelah praktikum ini, praktikan diharapkan dapat:

1. Memahami konsep Stack
2. Memahami operasi dasar Stack
3. Menerapkan Stack menggunakan bahasa C++

B. Peralatan

1. PC Desktop
2. Windows 7
3. MinGW

C. Teori

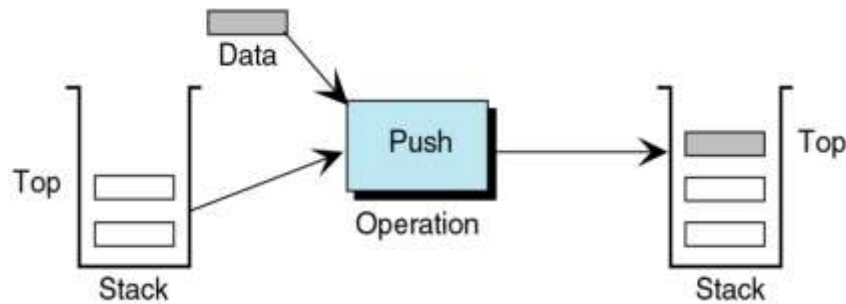
STACK (TUMPUKAN)

Stack nama lainnya tumpukan adalah Sekumpulan data yang diletakkan diatas data yan lain , dan bisa melakukan penambahan (penyisipan) data dan pengambilan (penghapusan) data lewat ujung yang sama. Konsep utamanya adalah LIFO (Last In First Out), benda yang terakhir masuk dalam stack akan menjadi benda yang dikeluarkan dari stack.

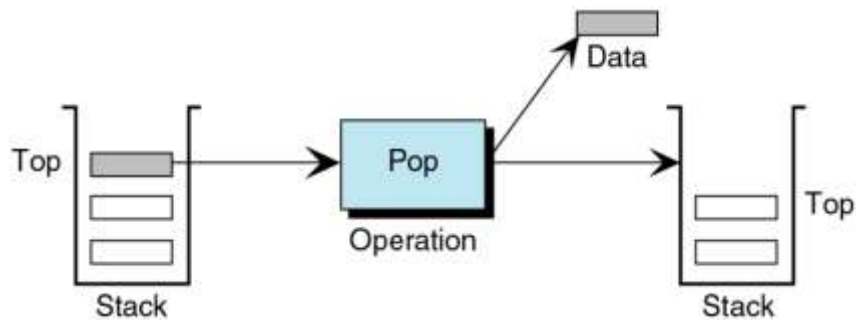


Gambar 1. Ilustrasi Stack

Terdapat 2 (dua) operasi yang dapat dikerjakan dalam stack : push dan pop. Push adalah operasi menambahkan data ke tumpukan, sedangkan pop adalah operasi mengambil data yang menempati posisi teratas dari stack.



Gambar 2. Operasi push pada stack



Gambar 3. Operasi pop pada stack

Operasi-operasi yang dapat dilakukan pada stack adalah :

1. Push (menambah) : operasi ini digunakan untuk menambahkan data ke dalam stack.
2. Pop (mengambil) : operasi ini digunakan untuk mengambil data yang ada di dalam stack.
3. IsFull (mengecek apakah stack penuh) : operasi ini digunakan untuk melakukan pengecekan apakah isi stack penuh atau tidak.
4. IsEmpty (mengecek apakah stack kosong) : operasi ini digunakan untuk melakukan pengecekan apakah isi stack kosong atau tidak.
5. Clear (membersihkan stack) : operasi ini digunakan untuk mengosongkan isi stack.
6. Print (mencetak isi stackprint) : operasi ini digunakan untuk menampilkan isi dari stack.

Langkah-langkah yang dilakukan dalam pembuatan stack :

1. Mendefinisikan max_stack untuk maksimum isi stack

```
#define max 10
```

2. Membuat deklarasi stack.

Stack dapat dideklarasikan dengan sebuah record yang mempunyai elemen sebuah array data untuk menyimpan elemen stack dan sebuah variabel atas untuk menyimpan elemen stack teratas (top elemen). Deklarasi selengkapnya sebagai berikut :

```
struct Tumpukan
{
    int atas;
    int data[max];
} T;
```

Keterangan :

Tumpukan adalah nama tipe pengenal untuk stack yang berbentuk record/struct.

int data[max] menunjukkan bahwa jumlah maksimal data yang dapat ditampung oleh stack tersebut berjumlah sebanyak [max].

3. Inisialisasi

Inisialisasi adalah proses untuk membuat stack dalam keadaan kosong. Proses ini dilakukan dengan mengisi variabel atas dengan nilai yang tidak menunjuk salah satu elemen array.

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

4. Operasi untuk melakukan pengecekan apakah data yang ada dalam stack kosong atau tidak. Apabila variabel atas menunjuk pada posisi -1 maka nilai yang akan dikembalikan adalah 1 (true) artinya stack dalam keadaan kosong. Namun jika variabel atas menunjuk pada posisi selain -1 maka nilai yang akan dikembalikan adalah 0 (false) yang artinya stack tidak dalam keadaan kosong.

```

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

```

5. Operasi untuk melakukan pengecekan apakah stack dalam keadaan penuh atau tidak. Apabila variabel atas menunjuk pada posisi max-1 maka nilai yang akan dikembalikan adalah 1 (true) artinya stack dalam keadaan penuh. Namun jika variabel atas menunjuk pada posisi selain max-1 maka nilai yang akan dikembalikan adalah 0 (false) yang artinya stack tidak dalam keadaan penuh.

```

int penuh()
{
    if(T.atas==max-1)
        return 1;
    else
        return 0;
}

```

6. Prosedur push (menginputkan data ke dalam stack). Sebelum menginputkan data, akan dilakukan pengecekan terlebih dahulu kondisi data yang ada di dalam stack. Jika stack dalam keadaan penuh maka data baru tidak dapat diinputkan ke dalam stack. Sebaliknya, jika stack tidak dalam keadaan penuh atau keadaan kosong maka bisa ditambahkan data ke dalam stack tersebut.

```

void input(int 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";
}

```

7. Prosedur pop (pengambilan data teratas dari stack). Prosesnya meliputi mengambil data dari elemen paling atas, kemudian menghapus elemen tersebut dengan cara mengubah nilai atas untuk menunjuk elemen dibawahnya.

```

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

```

D.I Praktikum

Latihan 1.1

Program lengkap dari contoh penerapan stack adalah sebagai berikut :

```
#include <iostream>
#include <cstdlib>
#define max 10

using namespace std;

struct Tumpukan
{
    int atas;
    int data[max];
} T;

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

int kosong()
{
    if(T.atas==max-1)
        return 1;
    else
        return 0;
}

int penuh()
{
    if(T.atas==max-1)
        return 1;
    else
        return 0;
}

void input(int 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";
}

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

void tampil()
{
    if(kosong()==0)
    {
        for(int i=T.atas;i>=0;i--)
        {
            cout<<"\nTumpukan ke " <<i<<"=" <<T.data[i];
        }
    }
    else
        cout<<"Tumpukan kosong";
}

void bersih()
{
    T.atas=-1;
    cout<<"Tumpukan kosong!";
}

```



```

int main()
{
    int pil,data;
    awal();
    do
    {
        cout<<"1. Input\n2. Hapus\n3. Tampil\n4. ";
        cout<<"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";
        }
        getchar();
        cout<<endl;
        cout<<endl;
    }
    while(pil!=5);
}

```

Latiham 1.2

```
using namespace std;

struct stack{
    int top;
    int tmp[max];
}stack;

void awal(){
    stack.top = -1 ;
}

int isEmpty(){
    if(stack.top == -1)
        return 1;
    else
        return 0;
}

int isFull(){
    if(stack.top == max - 1)
        return 1;
    else
        return 0;
}

void push (int data){
    if (isEmpty() == 1){
        stack.top++;
        stack.tmp[stack.top] = data;
        cout<<"Data "<<stack.tmp[stack.top]<<" Telah diinputkan";
    }
    else if (isFull() == 0){
        stack.top++;
        stack.tmp[stack.top] = data;
        cout<<"Data "<<stack.tmp[stack.top]<<" Telah masuk ke dalam Stack";
    }
    else{
        cout<<"Tumpukan Dalam Stack Penuh";
    }
}
```

```

void pop (){
    if(isEmpty() == 0){
        cout<<"Data Paling Atas Sudah Terambil";
        stack.top--;
    }
    else{
        cout<<"Data Dalam Stack Kosong";
    }
}

void print (){
    if(isEmpty() == 0){
        for (int i = stack.top; i>=0; i--){
            cout<<stack.tmp[i]<<" ";
        }
    }
    else{
        cout<<"Tumpukan Dalam Stack Kosong";
    }
}

```

```

main(){
    int pilih, data;
    awal();
    do{
        system("cls");
        cout<<"==== MENU PROGRAM STACK =====<<endl;
        cout<<"1. PUSH"<<endl;
        cout<<"2. POP"<<endl;
        cout<<"3. PRINT"<<endl;
        cout<<"4. EXIT"<<endl;
        cout<<"Masukan Pilihan Anda \t\t="<<endl;
        cin>>pilih;

        switch(pilih){
            case 1:
                cout<<"Masukan Data Ke Dalam Stack \t="<<endl;
                cin>>data;
                push(data);
                break;
            case 2:
                pop();
                break;
            case 3:
                print();
                break;
            case 4:
                cout<<"Tekan enter untuk keluar"<<endl;
                break;
        }
        getch();
    }
    while(pilih!=4);
}

```

Latiham 1.3

Program penerapan stack dengan pointer adalah sebagai berikut :

```
#include<iostream>
#include<stdio.h>
#include<stdlib.h>
#define max 10

using namespace std;

void push(int stack[],int *top, int value);
void pop(int stack[], int *top, int * value);
int main()
{
    int stack[max];
    int top=-1;
    int n, value;

    do
    {
        do
        {
            cout<<"Masukkan nilai yang akan di PUSH : ";
            cin>>value;
            push(stack,&top,value);

            cout<<"Tekan 1 untuk melanjutkan"<<endl;
            cin>>n;
        }while (n==1);
    }
```

```

cout<<"Tekan 1 untuk melanjutkan POP"<<endl;
cin>>n;

while (n==1)
{
    pop(stack,&top,&value);
    cout<<"Nilai yang di POP : "<<value<<endl;
    cout<<"Tekan 1 untuk melakukan POP sebuah elemen"<<endl;
    cin>>n;
}

cout<<endl;
cout<<"Tekan 1 untuk melanjutkan"<<endl;
cin>>n;
} while (n==1);
}

void push(int stack[],int *top, int value)
{
    if (*top < max)
    {
        *top = *top + 1;
        stack[*top] = value;
    }
    else
    {
        cout<<"Stack penuh, PUSH nilai tidak apat dilakukan"<<endl;
        exit(0);
    }
}
}

```

```

void pop (int stack[], int *top, int * value)
{
    if (*top>=0)
    {
        *value = stack[*top];
        *top=*top-1;
    }
    else
    {
        cout<<"Stack kosong, POP tidak dapat dilakukan"<<endl;
        exit(0);
    }
}

```

Output :

```

Masukkan nilai yang akan di PUSH : 1
Tekan 1 untuk melanjutkan
1
Masukkan nilai yang akan di PUSH : 2
Tekan 1 untuk melanjutkan
1
Masukkan nilai yang akan di PUSH : 3
Tekan 1 untuk melanjutkan
1
Masukkan nilai yang akan di PUSH : 4
Tekan 1 untuk melanjutkan
0
Tekan 1 untuk melanjutkan POP
1
Nilai yang di POP : 4
Tekan 1 untuk melakukan POP sebuah elemen
1
Nilai yang di POP : 3
Tekan 1 untuk melakukan POP sebuah elemen
1
Nilai yang di POP : 2
Tekan 1 untuk melakukan POP sebuah elemen
1
Nilai yang di POP : 1
Tekan 1 untuk melakukan POP sebuah elemen
1
Stack kosong, POP tidak dapat dilakukan

```

E. Tugas

1. Dengan menggunakan fungsi `push()` dan `pop()` pada stack, buatlah program untuk membalik kalimat.