PERTEMUAN 11
STACK

A. TUJUAN PEMBELAJARAN

Mampu menjelaskan pengertian dan pembuatan stack,melakukan operasi penyisipan dan penghapusan elemen dalam stack, dapat mengimplementasikan stack pada *array* dengan bahasa pemrograman *C++*. Dimodul ini anda harus mampu:

Ketepatan dan penguasaan dalam penggunaan konsep Stack pada *array* dalam membangun aplikasi dengan bahasa pemrograman *C++,* Latihan Dan tugas.

B. URAIAN MATERI

1. STACK

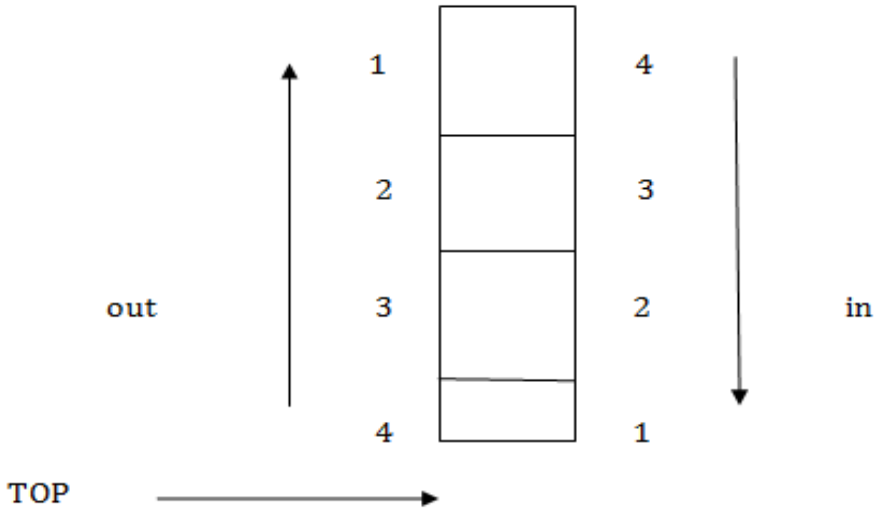
Stack atau tumpukan adalah kumpulan elemen yang hanya dapat di tambahatau dihapus dari satu ujung (gerbang) yang sama. Hal ini menunjukan bahwa seolah-olah suatu elemen diletakan di atas elemen yang lain. Yang memberi gambaran bahwa Stack mempunyai sifat LIFO(Last In First Out) yang berarti bahwa elemen yang terakhir masuk akan pertama keluar.

Stack dapat diartikan juga yaitu suatu struktur data linear, tehknik struktur data ini dimana data yang dimasukan paling akhir adalah data yang paling awal di keluarkan.

Secara sederhana stack dimisalkan kita mempunyai 4 buah kotak (A,B,C, dan D) yang ditumpukkan. Kotak A diletakkan paling bawah, lalu diikuti kotak B, C, dan yang teratas atau terakhir adalah D. Maka untuk mengambil tiap kotak harus dilakukan berurutan dari kotak D, C, B kemudian A. Karena jika kita mengambil kotak B tanpa terlebih dahulu mengambil kotak di atasnya maka tumpukan akan roboh.

Contoh sederhana selanjutnya, dimana suatu kelar belajar seorang siswa yang mengumpulkan tugas di paling akhir akan di koreksi di paling pertama.

Stack memiliki tipe data yang berifat abstrak namun seringkali bahkan sangat umum penggunaan stack dalam suatu pembuatan pemrograman.



Gambar 11.23 contoh stack LIFO :

a. Proses:
   1) awal (inisialisasi).
   2) Pop (hapus.mengambil.keluar)
   3) Push (input, simpan. Masuk)

b. Operasi dan fungsi Stack
   1) Push (memasukan) operasi ini digunakan menambah elemen di dalam stack, jika stack tidak penuh.
   2) Pop (keluar) operasi ini digunakan mengambil sekaligus menghapus elemen di dalam stack. Jika stack tidak kosong.
   3) IsEmpty adalah suatu fungsi cek stack sudah dalam keadaan kosong
   4) IsFul adalah suatu fungsi cek stack sudah dalam keadaan penuh.
   5) Display adalah fungsi menampilkan input program stack.

```
#include<iostream>

#define MAX_10

Using namespace std;
```

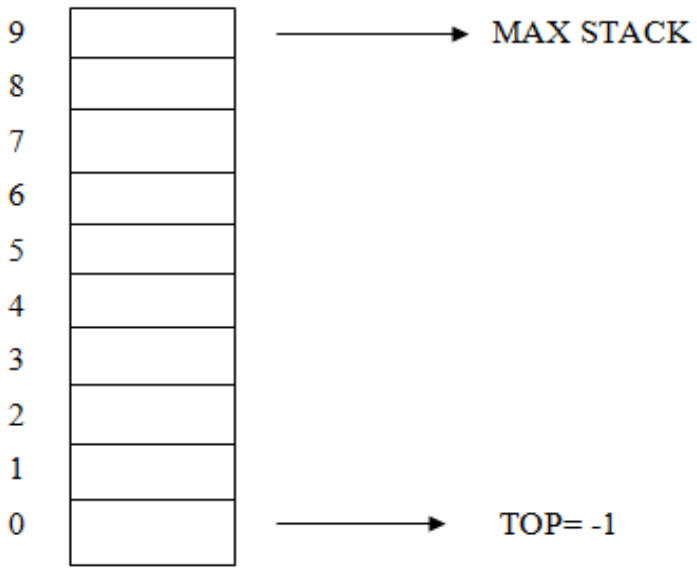
<iostream>header biasa digunakan dalam standar bahasa C++. Dan MAX untuk penggunaan data array dengan tumpukan (stack).

2. Implementasi Stack Dengan Array

Sturtuktur data untuk stack.

```
Struck tumpuk
{
        Int data[10]; //MAX 10 elemen di dalam stack
        Int top;//top of stack
};
 struct tumpuk;
```

TOP menunjukan elemen paling atas pada stack dan data array[10], bersamaan jumlah array data yaitu MAX.



Gambar 11.24 stack inisialisasi.

3. Deklarasi Inisialisai Top

a. Deklarasi inisialisasi Top.

```
int inisialisasi ()
{
                    Tumpuk.top= -1
};
```

Pastikan posisi Top sama dengan posisi 0 pada awal penambahan elemen, lalu masukan -1, karena array dalam C dimulai dari 0, yang berarti tumpuksn kodong.

Dan Top adalah variabel tumpukan yang mengarah elemen teratas Stack sekarang. TOS (Top Of Stack) akan bergerak sampai MAX of STACK sampai tumpukan penuh.

b. Deklarasi Max Stack

```
#Define max=10; //maximum 10 elemen stack
Tumpuk.top=-1// menunjukan bahwa tumpukan kosong
```

c. Deklarasi Cek Stack

```
Int IsEmpty()
{
Return tumpuk.Top== -1;
}
                    Int IsFull()
                    {
                    Return tumpuk.Top== Max-1;
                    }
```

1) fungsi IsEmpty : apabila nilai tumpukan.Top dengan -1 sama,maka true, apabila tidak, false.
2) fungsi IsFull : apabila nilai tumpukan.Top dengan Max-1 sama, maka true, apabila tidak, false.

d. STACK isEmpty

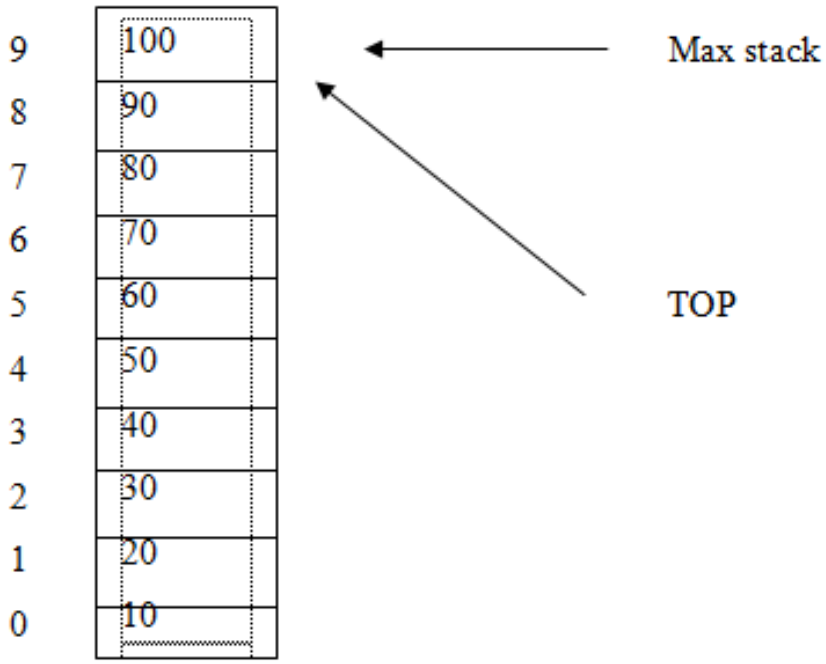
Contoh deklarasi Stack IsEmpty.

```
Int IsEmpty()
{
        If(Tumpuk.Top==1)) ; //jika TOP sama dengan null
        {
          Printf("Tumpuk Kosong") ;
          Return 1;

        }
        Else
          {
        Return 0;
          }
        }
```

1) Cek tumpukan kosong atau tidak.
2) Apabila kosong, tampilkan stack kosong.
3) Apabila ada, tampilkan tumpukan satu persatu dengan pengulangan.

e. Fungsi stack IsFull.

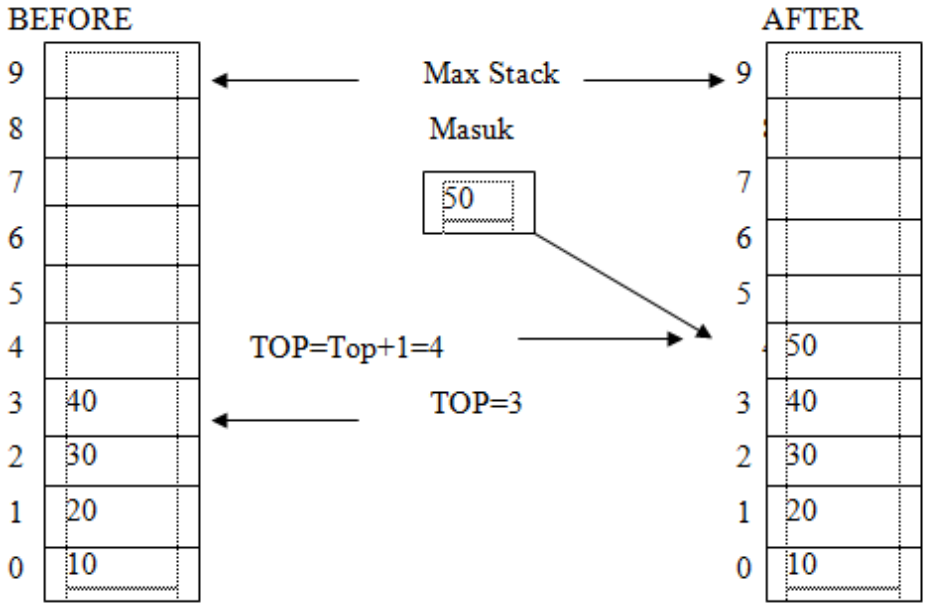


Gambar 11.25 stack IsFull.

Contoh deklarasi stack IsFull:

```
Int IsFull ()
        {
        if(Tumpuk.Top== max tumpuk-1); // jika TOP sama dengan posisi Max
        {
        Printf("Tumpuk Penuh");
        Return 1;
        Else
        Return 0;
        }
}
```

4. Operasi Push



Gambar 11.26 push.

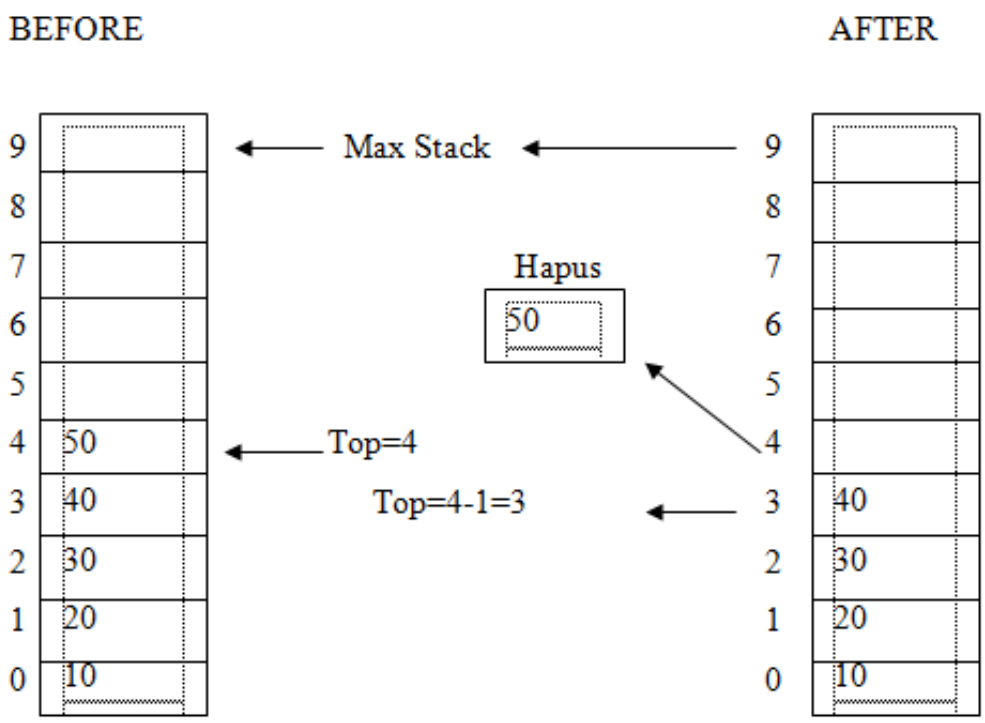
a. Cek Stack penuh atau tidak.
b. Apabila Stack penuh, tampilkan “Stack Penuh”
c. Apabila tidak penuh, tambah nilai (increment) diatas Stack Top untuk mengarah ke ruang lain.
d. Tambah elemen atau data yang mengarah Top.

Contoh deklarasi operasi Push.

```
Int Push (int d [10])
    {/* data elemen masuk kedalam stack. Ans menunggu elemen jika stack dalam keadaan full atau empty.
Int ans;
Ans=tumpuk.IsFull();
If (ans==0)//tumpuk tidak penuh.
        Tumpuk.top++; // increment stack pointer mengarah ke posisi
```

```
                                    selanjutnya
                tumpuk.data[tumpuk.top]=val);
        }
        else
        {
        Printf ("tumpuk penuh")
        }
        Return ();
        }
```

5. Operasi Pop



Gambar 11.27 Pop.

Operasi ini mengambil atau menghapus data atau elemen dari stack.

a. Cek apakah stack kosong atau tidak.
b. Apabila kosong, tampilkan “Stack Kosong”.
c. Apabila ada, tampilkan nilai teratas stack Top, lalu *decrement* nilai Top sampai berkurang elemen stack.

Contoh deklarasi operasi Pop.

```
Int pop();
{
Int ans;
Ans=tumpuk.IsEmpty();
If(ans==0)//tumpuk tidak kosong
{
   Printf(“elemen yang terambil=&s\n”,tumpuk data[tumpuk.top]);
   Tumpuk.top--;// decrement stack[Top]
}
Else
{
   Printf(“\n tumpuk kosong”);
}
```

6. Operasi Display

Fungsi ini menampilkan stack dari atas Top sampai Bottom structur LIFO.

Contoh deklarasi fungsi display atau tampilan.

```
Int Display()//fungsi display
{
```

```
Printf("\n");
If(tumpuk.IsEmpty()=0
        {       for(d==tumpuk.top;i--)// menampilkan item Stack
                Printf("%tampilkan\t"tumpuk.[i];
        }
        Else
                {
                        Printf("\tumpuk kosong");
                }
                Return 0;
        }
```

2.10 Contoh tampilan struktur data Stack dengan Array.

Fungsi-fungsi di atas setelah di deklarasikan, berlanjut membuat contoh menu menggunakan do while dengan ada switch didalam program.

Selanjutnya menciptakan menu tampilan yang terdapat pilihan menu *Push* lalu *Pop, Display*, dan fungsi-fungsi *IsEmpty* dan *IsFull.*

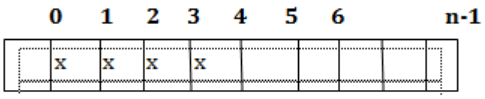
```
#include<iostream>
#define Max 10
Int push();
Int pop();
Int disp();
Int IsEmpty();
Int IsFull();
{
        Int data[10];
```

```
        Int Top;
};
Int main()
{
        Int pilih,item;
        tumpuk.Top=-1:
Do
{
        Printf("\n");
        Printf("\n menu");
        Printf("\n 1:push");
        Printf("\n 2:pop");
        Printf("\n 3:display")
        Printf("\n 4:exit")
        Printf("\n pilih :");
        Scanf("%d",&pilih);
{
        Case 1;
Printf("\n masukan elemen yang di hapus(pop) :");
Scanf("%d,&item);
                        Push(item);
                        Disp();
                        Break:
        Case 2;
                        Pop();
                        Disp();
                        Break;
```

```
        Case 3
                Disp;
                Break;
}
}while (pilih!=4);
Return 0;
//deklarasi fungsi
Int IsEmpty(){
If(tumpuk.Top==-1)//posisi Top sama dengan Null.
{
Printf("tumpuk Kosong");
Return 1;}
Else
        Return 0;
}
Int IsFull():{
If(stack.Top==max-1)//apabila posisi Top sama dengan Max.
{
Printf("Stack penuh"):
Return 1;
}
Else
Return 0;
}
Int push(val);{
Int ans=IsFull()
If(ans==0)//tumpuk belom penuh
```

```
{
tumpuk.Top++; //tambah nilai increment ke posisi selanjutnya
tumpuk.data[tumpuk.top]=val);
}
Else
{
Printf("tumpuk penuh");
}
Return 0;
}
Int pop();{
Int ans;
Ans=IsEmpty();
If(ans==0)//tumpuk tidak kosong
{
Printf("\n elemen pop adalah\t%d",tumpuk.data[tumpuk.top]);
Tumpuk.top--;//decrement tumpuk
}
Else
{
Printf("\n tumpuk kosong tidak ada item untuk di hapus");
}
Return 0;
}
```

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Tugas dan latihan : 	1. Buat algoritma dasar push dan pop? 2. Sebutkan ciri ciri stack kosong dan penuh? 3. Buat program input datasatu persatu dan push data sampai stack penuh? 4. Buat program hapus satu satu sampai stack kosong? 5. menambah increment nilai top setiap ada penambahan elemen stack selama belum penuh ,adalah langkah-langkah awal pada operasi stack?

D. REFERENSI

1.C And Data Structure by practice by Ramesh Vasappanvara.

2Implementasi Stack di C++ | Rahmat Subekti.

3Struktur Data Array Stack Dan Queue | Aries Indra Gunawan