**Stack**

Stack dapat diartikan sebagai tumpukan, stack sendiri di analogikan sebagai sebuah kotak yang akan diisi oleh beberapa barang. Stack bersifat**LIFO (Last In First Out)**artinya**data yang terakhir masuk merupakan data yang akan keluar terlebih dahulu.**

Operasi pada stack :

*Stack ( )* : untuk menginputkan suatu data.

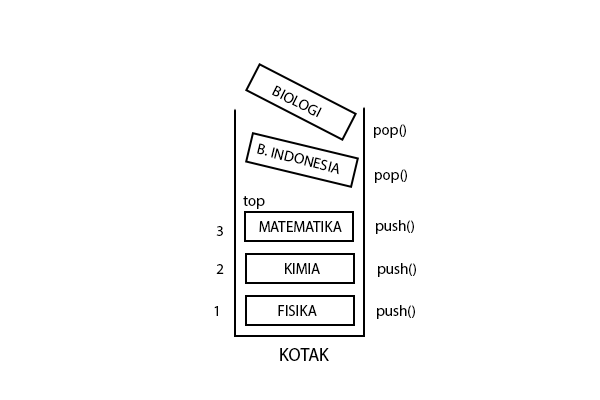
*Push (data)* : untuk memasukkan isi /bagian dari data.

*Pop ( )* : untuk mengeluarkan data terakhir (yang berada paling atas).

*Peek ( )* : untuk melihat data yang berada pada tumpukan paling atas (akan dikeluarkan).

*isEmpty ( )* : berfungsi untuk menguji apakah stack masih kosong.

stack dapat di ilustrasikan sebagai berikut :



dari ilustrasi di atas dapat diartikan dengan mengambil sifat LIFO, yaitu yang masuk, itulah yang akan keluar terlebih dahulu, karena biologi yang berada paling atas, maka yang akan keluar terlebih dahulu.

Jika dikaitkan dengan operasi pada stack, operasi diatas biologi di *pop* terlebih dahulu, kemudian dilanjut setelahnya, yaitu B Indonesia, begitupun seterusnya.

Berikut adalah contoh pemrograman menggunakan metode stack :

def stack():

s=[]

return (s)

def push (s,data):

s.append(data)

def pop(s):

data = s.pop()

return (data)

def peek (s):

return(s[len(s)-1])

def isEmpty(s):

return (s==[])

def size (s):

return(len(s))

def balik(teks):

a=stack()

hasil=''

for i in range(len(teks)):

push (a,teks[i])

for i in range(size(a)):

hasil=hasil+pop(a)

return hasil

print(balik('arek'))

**Queue.**

Queue pada Struktur Data atau antrian adalah sekumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung disebut dengan sisibelakang(rear), dan penghapusan(pengambilan elemen) dilakukan lewat ujung lain (disebut dengan sisi depan atau front).

Queue atau antrian prinsip yang digunakan adalah “Masuk Pertama Keluar Pertama” atau FIFO (First In First Out).

Queue atau antrian banyak kita jumpai dalam kehidupan sehari-hari, ex: antrian Mobil diloket Tol, Antrian mahasiswa Mendaftar, dll. Contoh lain dalam bidang komputer adalah pemakaian sistem komputer berbagi waktu(time-sharing computer system) dimana ada sejumlah pemakai yang akan menggunakan sistem tersebut secara serempak.

**Operasi Infix, Prefix, dan Postfix**

**Infix** adalah cara penulisan ungkapan dengan meletakkan operator di antara dua operand dalam hal ini pemakaian tanda kurung sangat menentukan hasil operasi. Contoh pemakaian infix adalah A+B, A+B-C,(A+B)\*(C-D), dll.

**Prefix** adalah metode penulisan dengan meletakkan operator di depan operand dan tanpa menuliskantandakurung. Contoh pemakaian prefix adalah  +AB, – +ABC, \* + AB – CD, dll.

**Postfix** adalah metode penulisan dengan menuliskan operator setelah operand dan tanpa menuliskan tanda kurung. Contoh pemakaian postfix adalah AB+,ABC+-, dll

Berikut adalah contoh program pengubahan dari operasi infix menjadi menjadi postfix :

def stack():

s=[]

return (s)

def push (s,data):

s.append(data)

def pop(s):

data = s.pop()

return (data)

def peek (s):

return(s[len(s)-1])

def isEmpty(s):

return (s==[])

def size (s):

return(len(s))

mathStr='A+B'

highPre='\*/'

lowPre='-+'

operator=highPre+lowPre

postStr=''

st=stack()

for i in mathStr:

if i in lowPre:

if isEmpty(st):

push(st,i)

else:

temp=peek(st)

#print('peek=',peek(st))

while (temp in operator) and not(isEmpty(st)) :

postStr=postStr+pop(st)

#print('while')

push(st,i)

elif i in highPre:

if isEmpty(st):

push(st,i)

else:

while not(isEmpty(st)) and (peek(st)in highPre):

postStr=postStr\*pop(st)

push(st,i)

elif i =='(':

push(st,i)

elif i==')':

while not(isEmpty(st)) and peek(st)!='(':

postStr=postStr+pop(st)

pop(st)

else:

postStr=postStr+i

print(postStr)

print('stack=',st)

while not(isEmpty(st)):

postStr=postStr+pop(st)

print(postStr)