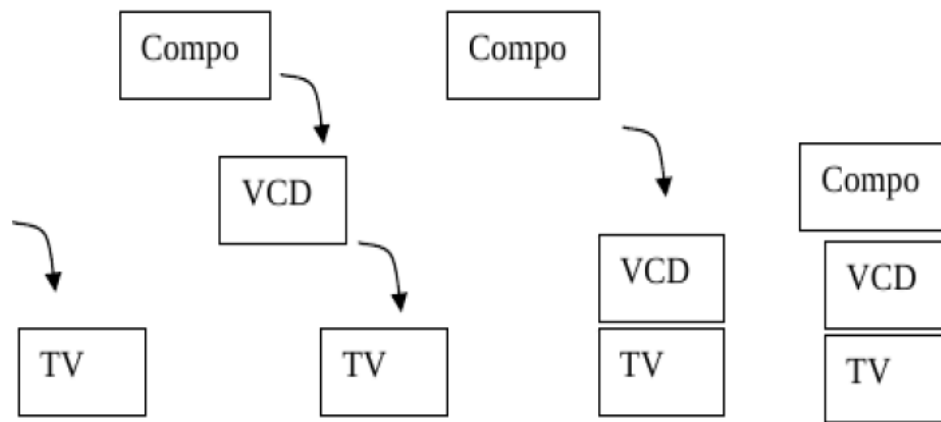


Minggu, 27 Juli 2019

LAPORAN KEGIATAN (STACK)

A. DASAR TEORI

1. Stack atau tumpukan adalah suatu struktur data yang penting dalam pemrograman. Stack bersifat LIFO (Last In First Out). Benda yang terakhir masuk ke dalam Stack akan menjadi benda pertama yang dikeluarkan dari stack. Contohnya, karena kita menumpuk Compo di posisi terakhir, maka Compo akan menjadi elemen teratas dalam tumpukan. Sebaliknya, karena kita menumpuk Televisi pada saat pertama kali, maka elemen Televisi menjadi elemen terbawah dari tumpukan. Dan jika kita mengambil elemen dari tumpukan, maka secara otomatis akan terambil elemen teratas, yaitu Compo juga.



B. PEMBAHASAN

1. Pada praktikum [pertama](#) ini akan mencoba membuat sebuah program dimana program ini telah jadi namun *source code* harus diubah menjadi ke dalam bahasa pemrograman Python 3.

```
1  class StackX:
2      max_size = 0
3      stack_array = []
4      top = 0
5
6      def __init__(self, s):
7          self.max_size = s
8          self.stack_array = [0] * s
9          self.top = -1
10
11      def push(self, j):
12          self.top += 1
13          self.stack_array[self.top] = j
14          return self.stack_array
15
16      def pop(self):
17          dummy = self.stack_array[self.top]
18          self.top -= 1
19          return dummy
20
21      def peek(self):
22          return self.stack_array[self.top]
```

```

23
24     def is_empty(self):
25         return self.top == -1
26
27     def is_full(self):
28         return self.top == self.max_size-1
29
30 class StackApp:
31
32     def __init__(self, jumlah_stack):
33         the_stack = StackX(jumlah_stack)
34         the_stack.push(20)
35         the_stack.push(40)
36         the_stack.push(60)
37         the_stack.push(80)
38
39         while not the_stack.is_empty():
40             value = the_stack.pop()
41             print(value, end=" ")
42             print("")
43
44 my_stack = StackApp(10)

```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk membuat sebuah class dengan nama **StackX**.
- Pada baris nomor 2, berfungsi untuk menentukan ukuran maksimum pada Stack.
- Pada baris nomor 3, berfungsi untuk membuat sebuah List kosong.
- Pada baris nomor 4, berfungsi untuk mendapatkan indeks paling atas pada Stack.
- Pada baris nomor 6 sampai 9, berfungsi untuk insialisasi terhadap variabel **max_size**, **stack_array**, dan **top**.

- Pada baris nomor 11 sampai 14, berfungsi untuk memasukan elemen ke dalam Stack.
 - Pada baris nomor 16 sampai 19, berfungsi untuk menghilangkan elemen paling atas pada Stack.
 - Pada baris nomor 21 sampai 22, berfungsi untuk melihat elemen paling atas pada Stack.
 - Pada baris nomor 24 sampai 25, berfungsi untuk membuktikan apakah Stack tersebut kosong atau tidak.
 - Pada baris nomor 27 sampai 28, berfungsi untuk membuktikan apakah Stack tersebut sudah penuh atau belum.
 - Pada baris nomor 30, berfungsi untuk membuat sebuah class dengan nama **StackApp**.
 - Pada baris nomor 32 sampai 33, berfungsi untuk insialisasi sekaligus menerima parameter **jumlah_stack** dan membuat sebuah objek bernama **the_stack**. Kemudian parameter **jumlah_stack** akan di oper ke objek **the_stack**.
 - Pada baris nomor 34 sampai 37, berfungsi untuk memasukan elemen ke dalam Stack.
 - Pada baris nomor 67, berfungsi untuk menampilkan elemen-elemen yang ada di dalam Stack.
2. Pada praktikum [kedua](#) ini akan menjalankan program pada praktikum [pertama](#), kemudian menjelaskan hasil program tersebut.

```
80 60 40 20
>>> |
```

Output Program

* Penjelasan Output Program di atas

- a. Pada output diatas muncul 4 angka yaitu 80, 60, 40 dan 20. Dimana angka tersebut muncul dengan melewati beberapa tahap. Pertama angka-angka tersebut akan dimasukan ke dalam List kosong. Setelah itu angka paling atas akan dikeluarkan terlebih dahulu, hingga sampai angka paling bawah. Angka tadi bentuk semulanya adalah 20, 40, 60, dan 80. Dikarenakan angka 80 paling atas maka angka 80 akan keluar paling pertama. Sehingga outputnya adalah 80, 60, 40 dan 20.
3. Pada praktikum [ketiga](#) ini akan mengembangkan program pada praktikum sebelumnya dimana ada penambahan *function* yang berfungsi untuk menghapus salah satu elemen di dalam stack.

```
30     def delete_element(self, j):
31         self.stack_array.remove(j)
32         return j
```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 30, berfungsi membuat sebuah *function* dengan nama **delete_element** yang menerima parameter **j**. Parameter **j** adalah angka/element yang akan dihapus di dalam Stack.
 - Pada baris nomor 31, berfungsi untuk melakukan penghapusan terhadap elemen di dalam Stack dengan menggunakan fungsi `remove()`
4. Pada praktikum [keempat](#) ini akan mengembangkan program pada praktikum sebelumnya dimana ada penambahan *function* yang berfungsi untuk mencari suatu elemen di dalam Stack.

```
34     def search_element(self, j):
35         if j in self.stack_array:
36             print(str(j)+" Ditemukan")
37         else:
38             print(str(j)+" tidak ditemukan")
```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 34, berfungsi sebuah *function* dengan nama **search_element** yang menerima parameter **j**. Parameter **j** adalah angka/element yang ingin dicari di dalam Stack
 - Pada baris nomor 35 sampai 38, berfungsi untuk membuat suatu perbandingan dimana jika angka terdapat di dalam Stack maka akan menampilkan pesan angka ditemukan. Jika tidak maka akan menampilkan pesan angka tidak ditemukan.
5. Pada praktikum kelima ini akan mengembangkan program pada praktikum sebelumnya dimana ada penambahan *function* yang berfungsi untuk mengubah suatu elemen di dalam Stack.

```
40     def edit_element(self, j, k):
41         for i in range(len(self.stack_array)):
42             if self.stack_array[i] == j:
43                 self.stack_array[i] = k
44             return self.stack_array[i]
```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 40, berfungsi untuk
- Pada baris nomor 2, berfungsi membuat sebuah *function* dengan nama **edit_element** yang menerima parameter **j** dan **k**. Parameter **j** adalah angka/element yang ingin di edit sedangkan parameter **k** adalah angka yang mau di *replace*.
- Pada baris nomor 41, berfungsi untuk melakukan perulangan sebanyak jumlah elemen pada stack.
- Pada baris nomor 42 sampai 43, berfungsi untuk melakukan perbandingan jika angka yang ingin di edit terdapat pada Stack, maka selanjutnya akan dilakukan perubahan elemen dengan variabel **K**.

- Pada baris nomor 44, berfungsi untuk mengembalikan nilai dari hasil perubahan nilai.
6. Pada praktikum [keenam](#) ini akan mengembangkan program pada praktikum sebelumnya dimana ada penambahan *function*, masing-masing *function* berfungsi untuk mencari nilai total, rata-rata, nilai terbesar dan nilai terkecil di dalam Stack.

```
46     def cari_nilai_total(self):
47         return sum(self.stack_array)
48
49     def cari_nilai_rata_rata(self):
50         return sum(self.stack_array) / len(self.stack_array)
51
52     def cari_nilai_terbesar(self):
53         return max(self.stack_array)
54
55     def cari_nilai_terkecil(self):
56         return min(self.stack_array)
```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 46 sampai 47, berfungsi untuk mendapatkan total nilai pada Stack dengan menggunakan fungsi `sum()`
- Pada baris nomor 49 sampai 50, berfungsi untuk mendapatkan nilai rata rata pada Stack dengan menggunakan cara mentotalkan nilai dari Stack kemudian di bagi dengan jumlah nilai di dalam Stack.
- Pada baris nomor 52 sampai 53, berfungsi untuk mendapatkan nilai terbesar pada Stack dengan menggunakan fungsi `max()`.
- Pada baris nomor 55 sampai 56, berfungsi untuk mendapatkan nilai terkecil pada Stack dengan menggunakan fungsi `min()`.

C. KESIMPULAN

Stack atau tumpukan adalah struktur data yang memiliki sifat LIFO (Last In First Out) yang artinya semua benda ketika pertama kali masuk akan keluar paling terakhir. Di ibaratkan seperti kardus yang menyimpan banyak buku, ketika ingin mengambil buku yang letaknya tertimpa oleh buku lainnya maka buku yang menimpa tersebut akan dikeluarkan terlebih dahulu.

D. DAFTAR PUSTAKA

Wibowo,Fitri. __, *Stack*, [pdf]