

Queue

- a) Antrian, pengoperasian berdasarkan aturan FIFO (First in first out). sama modelnya dengan antrian, yang pertama mengantri dialah yang diambil. Kebalikan dari Stack.
- b) Terdapat satu ujung yang bernama “ekor” (rear) dan satu ujung yang bernama “depan” (front). penambahan data terjadi di ekor dan penghapusan dari depan.
- c) Operasi dalam Queue :
- d) Queue() = inisialisasi variabel yang mendandakan bahwa Queue
- e) Enqueue(item) = menambahkan item kedalam sebuah Queue
- f) Dequeue(item) = menghapus item dalam Queue
- g) isEmpty() = mengecek data
- h) Size() = mengetahui ukuran suatu Queue

Syntax konsep Queue :

```
def createQueue():  
    q = []  
    return (q)  
def enqueue(q,data):  
    q.insert(0,data)  
    return (q)  
def dequeue(q):  
    data = q.pop()  
    return (data)  
def isEmpty(q):  
    return (q==[])  
def size(q):  
    return (len(q))
```

Deque

- a) Antrian berujung dua, sama dengan Queue. Tapi untuk penambahan dan penghapusan data terjadi dimana saja, bisa di rear atau di front.
- b) Operasi dalam Queue :
- c) deque() = inisialisasi variabel yang mendandakan bahwa Queue
- d) addFront(item) = menambahkan item ke depan
- e) addRear(item) = menambahkan item ke rear
- f) removeFront() = menghapus data di front
- g) removeRear() = menghapus data di rear
- h) isEmpty() = mengecek data dalam deque
- i) Size() = mengetahui ukuran suatu deque

Syntax konsep Deque :

```

def deque():
    d = []
    return (d)
def addFront(d,data):
    d.append(data)
    return (d)
def addRear(d,data):
    d.insert(0,data)
    return (d)
def removeRear(d):
    data = d.pop(0)
    return (data)
def removeFront(d):
    data = d.pop()
    return (data)
def isEmpty(d):
    return (d==[])
def size(d):
    return (len(d))

```

LINKED LIST

Sejauh ini, kita sudah menggunakan Python lists untuk mengimplementasikan berbagai tipe data abstrak. List bersifat powerful namun simpel, mekanisme koleksi yang menyediakan programmer variasi operasi yang luas. Namun tidak semua bahasa pemrograman menyertakan suatu koleksi list. Pada kasus demikian, gagasan mengenai list harus diimplementasikan oleh programmer.

List adalah koleksi item dimana setiap item berposisi relatif terhadap yang lain. Ini disebut pula unordered list. List punya item pertama, kedua, ketiga dan seterusnya. Dapat pula dirujuk awal list (item pertama) atau ujung list (item terakhir).

```

class Node:
    def __init__(self,init_data):
        self.data = init_data
        self.next = None
    def getData(self):
        return self.data
    def getNext(self):
        return self.next
    def setData(self,newdata):
        self.data = newdata
    def setNext(self,new_next):
        self.next = new_next

class LinkedList:
    def __init__(self):
        self.head = None
    def isEmpty(self):
        return self.head == None
    def add (self,item):
        temp = Node(item)
        temp.setNext(self.head)

```

```

        self.head = temp
def size(self):
    current = self.head
    count = 0
    while current != None:
        count = count + 1
        current = current.getNext()
    return count
def display(self):
    current = self.head
    while current != None:
        print(current.getData())
        current = current.getNext()
def search(self,data):
    current = self.head
    found = False
    index = 0
    temp = 0
    while current != None and not found:
        index = index + 1
        if (current.getData() == data):
            temp = current.getData()
            found = True
        else:
            current = current.getNext()
    if (temp == data):
        return "data ke",index-1
    else:
        return "data tidak ada"
def delet(self,item):
    current = self.head
    pre = None
    found = False
    while not found and current != None:
        if current.getData() == item:
            found = True
        else:
            pre = current
            current = current.getNext()
    if current != None:
        if pre==None:
            self.head = current.getNext()
        else:
            pre.setNext(current.getNext())
        print("berhasil dihapus")
    else:
        print ("data tidak ada yang di hapus")

```