Deques

Indah Agustien Siradjuddin

Struktur Data

Struktur data yang ketiga adalah **Deques**. Jika pada stacks, penambahan data baru dan penghapusan dilakukan pada ujung yang sama, yaitu **top**, sedangkan pada queues, penambahan data baru dan penghapusan dilakukan pada ujung yang berbeda, yaitu **rear**(penambahan data) dan **front**(penghapusan data). Maka struktur data **deques** dapat dilakukan di kedua ujung, seperti halnya queues, hanya terdapat sedikit perbedaan.
penambahan data dapat dilakukan pada kedua ujung, baik **front** maupun **rear**. Begitu juga halnya dengan penghapusan data yang dapat dilakukan pada **front** maupun **rear**.

Berikut pembahasan pada struktur data **Deques**:

Operasi Deques

Operasi-operasi yang terdapat pada *Deques* antara lain :

```
- deque (),         inisialisasi struktur data deque kosong
- addFront (data),  penambahan data baru di ujung front pada deque
- addRear (data),   penambahan data baru di ujung rear pada deque
- removeRear (),    penghapusan data pada ujung rear
- removeFront (),   penghapusan data pada ujung front
- isEmpty(),        pengecekan apakah deque dalam keadaan kosong
- size (),          informasi jumlah data yang terdapat pada deque
```

Struktur data *Deques* ini direpresentasikan dengan *List*, dimana:

- indeks awal list, yaitu indeks ke-0, adalah posisi **front** pada deques
- indeks akhir list, yaitu indeks terakhir, adalah posisi **rear** pada deques

Oleh karena itu, penambahan data baru pada posisi **front** dapat dilakukan dengan method `insert()` pada posisi ke-0. Sedangkan penambahan data baru pada posisi **rear** dapat dilakukan dengan method `append`, yang merupakan penambahan data baru pada list di posisi terakhir.

Sedangkan penghapusan data pada posisi **front** dapat dilakukan dengan method `pop()` pada posisi ke-0, sedangkan penghapusan data pada posisi **rear** dilakukan dengan method `pop()`.

Code

Berikut adalam implementasi deques dengan menggunakan List :

In [1]:
```
def createDeque():
    d=[]
    return (d)
def addFront(d,data):
    d.insert(0,data)
    return(d)
def addRear(d,data):
    d.append(data)
    return(d)
def removeRear(d):
    data=d.pop()
    return(data)
def removeFront(d):
    data=d.pop(0)
    return(data)
def isEmpty(d):
    return (d==[])
def size(d):
    return (len(d))
```

Berikut ini adalah contoh penggunan deques

In [3]:
```
deq=createDeque()
```

In [4]:
```
addFront(deq,'struktur')
addFront(deq,'data')
```

Out[4]:
```
['data', 'struktur']
```

In [5]:
```
addRear(deq,'2018')
addRear(deq,'2019')
```

Out[5]:
```
['data', 'struktur', '2018', '2019']
```

In [6]:
```
addRear(deq,'100')
```

Out[6]:
```
['data', 'struktur', '2018', '2019', '100']
```

In [7]:
```
data=removeFront(deq)
print(data)
print(deq)
```

```
data
['struktur', '2018', '2019', '100']
```

In [8]:
```
data=removeRear(deq)
print(data)
print(deq)
```

```
100
['struktur', '2018', '2019']
```

In [9]:
```
size(deq)
```

Out[9]: 3

In [10]:
```
isEmpty(deq)
```

Out[10]: False

In [11]:
```
addRear(deq,removeFront(deq))
print(deq)
```

```
['2018', '2019', 'struktur']
```

[Kembali ke Menu Awal]

Implementasi Deques

Contoh implementasi Deques adalah pengecekan apakah suatu kata merupakan palindrom. Palindrom merupakan rangkaian kata atau bilangan yang terbaca sama baik dari depan maupun dari belakang.
Struktur data **deques** paling tepat digunakan untuk memeriksa apakah suatu kata maupun bilangan adalah palindrom. Untuk memeriksa apakah suatu kata adalah palindrome, dapat dilakukan dengan memeriksa apakah ujung yang terletak pada front dan ujung yang terletak pada rear, merupakan karakter yang sama.

Code

Berikut adalah code untuk pengecekan palindrom dari suatu kata

In [16]:
```
def cekPalindrom(string):
    palindrom=createDeque()
    for huruf in string:
        addRear(palindrom,huruf)
    cek=True
    while size(palindrom)>1:
        a=removeRear(palindrom)
        b=removeFront(palindrom)
        if (a==b):
            cek=cek and True
        else:
            cek=cek and False
    return cek
```

In [19]:
```
print(cekPalindrom('hannah'))
print(cekPalindrom('surabaya'))
print(cekPalindrom('abcdcba'))
print(cekPalindrom('katak'))
print(cekPalindrom('taat'))
print(cekPalindrom('dia'))
```

```
True
False
True
True
True
False
```

Kembali ke Menu Awal