Linked List

```
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, newNext):
        self.next = newNext


class linkedList:
    def __init__(self):
        self.head = None

    def isEmpty(self):
        return self.head == None

    def display(self):
        current = self.head
        while current != None:
            print(current.getData(), end=" ")
            current = current.getNext()
        print("\n")

    def addFront(self, item):
        temp = Node(item)
        temp.setNext(self.head)
        self.head = temp


    def addRear(self, item):
        temp = Node(item)
        current = self.head
        if current == None:
            temp.setNext(current)
            current = temp
        while current.getNext() != None:
            current = current.getNext()
        current.setNext(temp)

    def insertNext(self, target, item):
        current = self.head
        found = False

        while current != None and not found:
            if current.getData() == target:
                found = True
            else:
                current = current.getNext()

        if found == True:
            temp = Node(item)
            temp.setNext(current.getNext())
            current.setNext(temp)
        else:
            print("Target node tidak ditemukan dalam linked list.")

    def removeFront(self):
        if self.isEmpty():
            print("linked list kosong")
        else:
            self.head = self.head.getNext()
```

```
    def removeRear(self):
        if self.isEmpty():
            print("linked list kosong")
        else:
            current = self.head
            previous = None
            while current.getNext() != None:
                previous = current
                current = current.getNext()
            if previous == None:
                self.head = None
            else:
                previous.setNext(None)

    def removeNode(self, item):
        if self.isEmpty():
            print("linked list kosong")
        else:
            current = self.head
            previous = None
            found = False
            while not found:
                if current.getData() == item:
                    found = True
                else:
                    previous = current
                    current = current.getNext()
            if previous == None:
                self.head = current.getNext()
            else:
                previous.setNext(current.getNext())


myList = linkedList()


a = Node(400)
a.setNext(myList.head)
myList.head = a
b = Node(300)
b.setNext(myList.head)
myList.head = b
c = Node(200)
c.setNext(myList.head)
myList.head = c
d = Node(100)
d.setNext(myList.head)
myList.head = d
myList.display()
```

```
 100 200 300 400
```

menambahkan data 500 dari belakang

```
print("Penambahan data 500 dari belakang")
myList.addRear(500)
myList.display()
```

```
 Penambahan data 500 dari belakang
 100 200 300 400 500
```

menambahkan data 50 dari depan

```
print("penambahan data 50 dari depan")
myList.addFront(50)
myList.display()
```

```
 penambahan data 50 dari depan
 50 100 200 300 400 500
```

menambahkan data 250 setelah node 200

```
print("penambahan data 250 setelah  node 200")
myList.insertNext(200, 250)
myList.display()
```

```
penambahan data 250 setelah  node 200
50 100 200 250 300 400 500
```

menghapus node depan, belakang dan menghapus node yang memiliki data 300

```
print("hapus node depan, node belakang, node yang memiliki data 300")
myList.removeFront()
myList.removeRear()
myList.removeNode(300)
myList.display()
```

```
hapus node depan, node belakang, node yang memiliki data 300
100 200 250 400
```