**SOAL :**

1. Buatlah program double linked list non circular dengan menggunakan head, namun head yang ada tidak digunakan untuk menyimpan data, melainkan untuk menyimpan jumlah seluruh elemen dalam linked list.
2. Buatlah fungsi lengkap untuk tambah, hapus, lihat, dan edit!
3. Buatlah pula function untuk menampilkan data list secara terbalik!

**JAWABAN :**

*Source Code :*

print('========================')

print('Nama : Febro Herdyanto')

print('NIM : 312010043')

print('Kelas : TI.20.B.1')

print('Task : Double Linked List Non Circular')

print('========================')

class Node:

def \_\_init\_\_(self, initdata):

self.data = initdata

self.next = None

self.previous = None

def getData(self):

return self.data

def getNext(self):

return self.next

def getPrevious(self):

return self.previous

def setData(self, newdata):

self.data = newdata

def setNext(self, newnext):

self.next = newnext

def setPrevious(self, newprevious):

self.previous = newprevious

class Unorderedlist:

def \_\_init\_\_(self):

self.head = None

def show(self):

current = self.head

print('None <-')

print('Head ->', end='')

while current != None:

if current.getNext() == None:

print(current.getData(), end='->')

else:

print(current.getData(), end='<->')

current = current.getNext()

print('None')

def isEmpty(self):

return self.head == None

def add(self, item):

temp = Node(item)

temp.setNext(self.head)

temp.setPrevious(None)

self.head = temp

def size(self):

current = self.head

count = 0

while current != None:

count += 1

current = current.getNext()

return count

def search(self, item):

current = self.head

found = False

while current != None and not found:

if current.getData() == item:

found = True

else:

current = current.getNext()

return found

def remove(self, item):

current = self.head

previous = None

found = False

while current != None and not found:

if current.getData() == item:

found = True

else:

previous = current

current = current.getNext()

if current == None:

next = None

else:

next = current.getNext()

if previous == None:

self.head = current.getNext()

else:

if current == None:

previous.setNext(None)

else:

previous.setNext(current.getNext())

if next != None:

next.setPrevious(current.getPrevious)

if current is None:

print('Data not in list')

mylist = Unorderedlist()

mylist.add(31)

mylist.add(11)

mylist.add(15)

mylist.add(30)

mylist.show()

def remove(self, item):

current = self.head

previous = None

found = False

while current != None and not found:

if current.getData() == item:

found = True

else:

previous = current

current = current.getNext()

if current == None:

next = None

else:

next = current.getNext()

if previous == None:

self.head = current.getNext()

else:

if current == None:

previous.setNext(None)

else:

previous.setNext(current.getNext())

if next != None:

next.setPrevious(current.getPrevious)

if current is None:

print('Data not in list')

mylist = Unorderedlist()

mylist.add(31)

mylist.add(11)

mylist.add(15)

mylist.add(30)

mylist.show()

Text

Description automatically generated