#SINGLE LINKED LIST

#=======================================================================

import os

# Membuat class untuk node

class Node(object):

def \_\_init\_\_(self, data=None, next\_node=None):

self.data = data

self.next\_node = next\_node

# Mengambil data dari node

def get\_data(self):

return self.data

# Mengambil node berikutnya

def get\_next(self):

return self.next\_node

# Menentukan node berikutnya

def set\_next(self, new\_next):

self.next\_node = new\_next

# Membuat class untuk linked list

class LinkedList(object):

def \_\_init\_\_(self, head=None):

self.head = head

# Menambah node baru

def insert(self, data):

# Inisialisasi node baru

new\_node = Node(data)

# Menunjuk node berikutnya dari node baru ke node yang ditunjuk oleh HEAD

new\_node.set\_next(self.head)

# HEAD menunjuk ke node baru

self.head = new\_node

# Menghitung panjang list

def size(self):

# Membuat pointer baru menunjuk ke node yang ditunjuk oleh HEAD

current = self.head

count = 0

# Perulangan untuk menghitung node

while current:

count += 1

current = current.get\_next()

return count

# Mencari sebuah data pada list

def search(self, data):

# Membuat pointer baru menunjuk ke node yang ditunjuk oleh HEAD

current = self.head

found = False

# Perulangan mencari node yang dicari

while current and found is False:

if current.get\_data() == data:

found = True

else:

current = current.get\_next()

return found

# Menghapus node

def delete(self, data):

current = self.head

previous = None

found = False

while current and found is False:

if current.get\_data() == data:

found = True

else:

previous = current

current = current.get\_next()

if current is None:

raise ValueError("Data not in list")

if previous is None:

self.head = current.get\_next()

else:

previous.set\_next(current.get\_next())

# Menampilkan isi dari list

def showData(self):

os.system('clear')

print ("Tampilkan list data:")

print ("Node -> Next Node")

current\_node = self.head

while current\_node is not None:

print (current\_node.data),

print (" ->"),

print (current\_node.next\_node.data) if hasattr(current\_node.next\_node, "data") else None

current\_node = current\_node.next\_node

# Main menu aplikasi

def mainmenu(self):

pilih = "y"

while (pilih == "y"):

os.system("clear")

print("===============================")

print("| Menu aplikasi linked list |")

print("===============================")

print("1. Insert data")

print("2. Delete data")

print("3. Cari data")

print("4. Panjang dari linked list")

print("5. Tampil data")

print("===============================")

pilihan=str(input(("Silakan masukan pilihan anda: ")))

if(pilihan=="1"):

os.system("clear")

obj = str(input("Masukan data yang ingin anda tambahkan: "))

self.insert(obj)

elif(pilihan=="2"):

os.system("clear")

obj = str(input("Masukan data yang ingin anda dihapus: "))

self.delete(obj)

x = input("")

elif(pilihan=="3"):

os.system("clear")

obj = str(input("Masukan data yang ingin anda dicari: "))

status = self.search(obj)

if status == True:

print("Data ditemukan pada list")

else:

print("Data tidak ditemukan")

x = input("")

elif(pilihan=="4"):

os.system("clear")

print("Panjang dari queue adalah: "+str(self.size()))

x = input("")

elif(pilihan=="5"):

os.system("clear")

self.showData()

x = input("")

else:

pilih="n"

if \_\_name\_\_ == "\_\_main\_\_":

# execute only if run as a script

l = LinkedList()

l.mainmenu()