**SOAL:**

1. Buatlah program lengkap dari semua algoritma dan function di atas dalam bentuk menu untuk menambah data, melihat data, dan menghapus data!
2. Buatlah function tambahan yang berguna untuk mencari data yang ada dalam linked list baik secara ber-Head maupun ber-Head dan Tail!
3. Buatlah function untuk menghapus data tertentu dalam linked list!
4. Buatlah penyisipan node setelah atau sebelum data tertentu.

**JAWABAN:**

*Source Code Programs :*

import os

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

print('Nama : Febro Herdyanto')

print('NIM : 312010043')

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

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

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

class Node(object):

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

self.data = data

self.next\_node = next\_node

def get\_data(self):

return self.data

def get\_next(self):

return self.next\_node

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

self.next\_node = new\_next

class LinkedList(object):

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

self.head = head

def insert(self, data):

new\_node = Node(data)

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

self.head = new\_node

def size(self):

current = self.head

count = 0

while current:

count += 1

current = current.get\_next()

return count

def search(self, data):

current = self.head

found = False

while current and found is False:

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

found = True

else:

current = current.get\_next()

return found

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())

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

def mainmenu(self):

pilih = "y"

while pilih == "y":

os.system("clear")

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

print("| Menu aplikasi linked list |")

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

print("1. Tambah data")

print("2. Hapus data")

print("3. Cari data")

print("4. Lihat jumlah data")

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("Jumlah data adalah: " + str(self.size()))

x = input("")

elif (pilihan == "5"):

os.system("clear")

self.showData()

x = input("")

else:

pilih = "n"

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

l = LinkedList()

l.mainmenu()

***Screenshot Programs :***

Text

Description automatically generated

Text

Description automatically generated

A picture containing shape

Description automatically generated

Shape

Description automatically generated with medium confidence

A picture containing shape

Description automatically generated

Text

Description automatically generatedShape

Description automatically generated with medium confidence