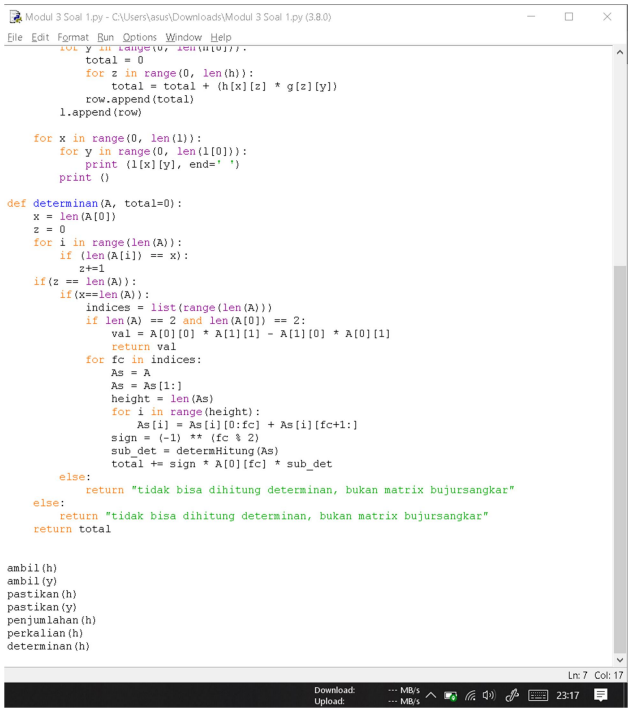
Nama : Riyan Sutantio Bangkit N.

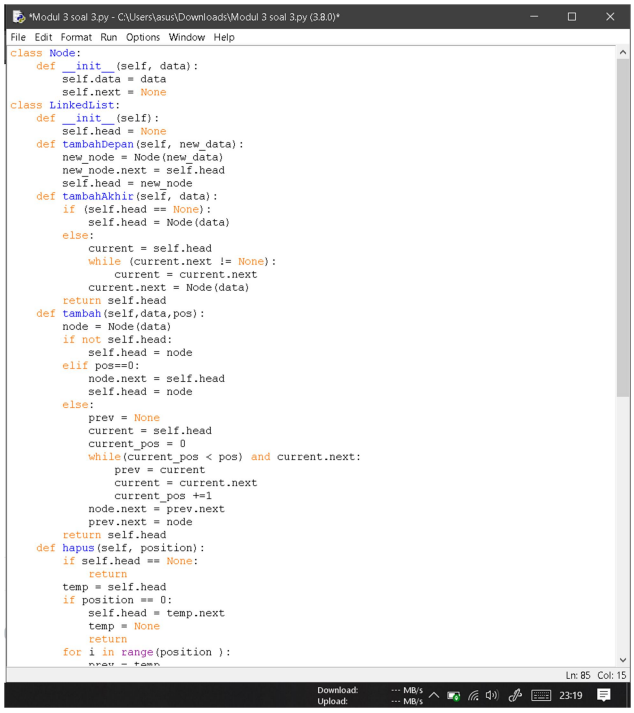
NIM : L200180180

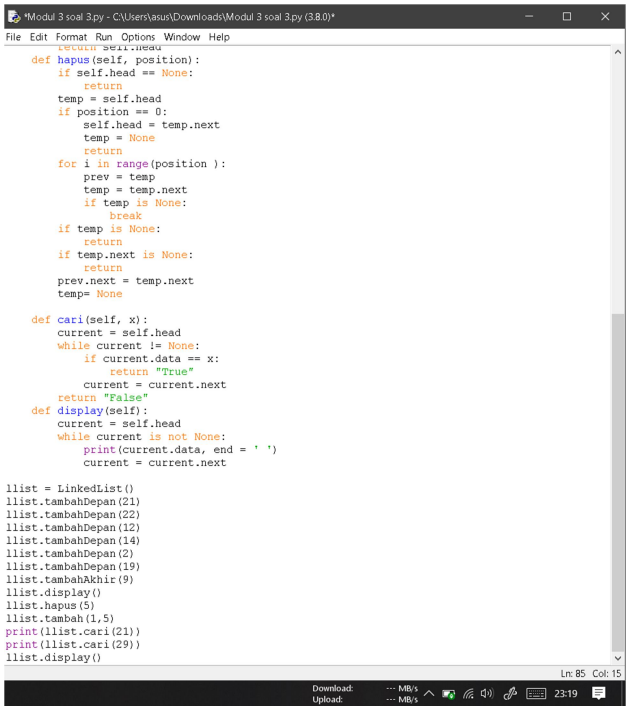
Modul 3

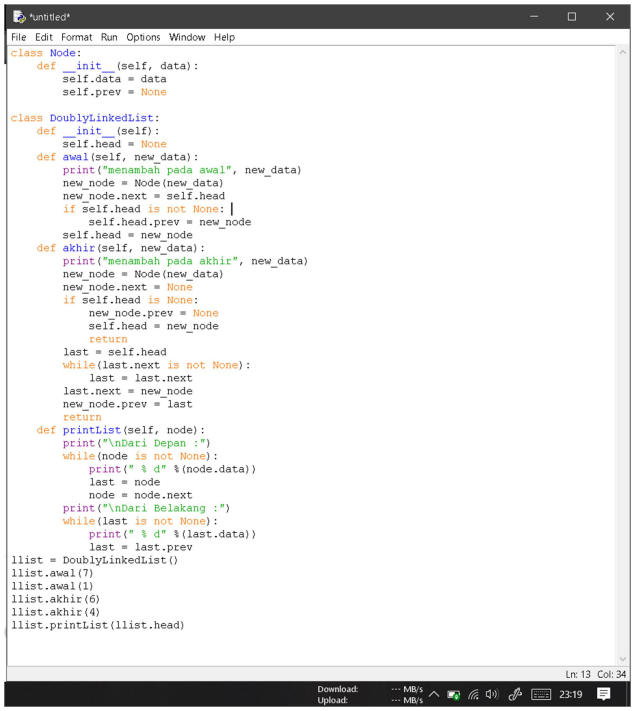
Algoritma dan Struktur Data


Modul 3 Soal 1.py - C:\Users\asus\Downloads\Modul 3 Soal 1.py (3.8.0)
File Edit Format Run Options Window Help
for y in range(0, len(h[0])):
total = 0
for z in range(0, len(h)):
total = total + (h[x][z] * g[z][y])
row.append(total)
l.append(row)
for x in range(0, len(l)):
for y in range(0, len(l[0])):
print (l[x][y], end=' ')
print ()
def determinan(A, total=0):
x = len(A[0])
z = 0
for i in range(len(A)):
if (len(A[i]) == x):
z+=1
if(z == len(A)):
if(x==len(A)):
indices = list(range(len(A)))
if len(A) == 2 and len(A[0]) == 2:
val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
return val
for fc in indices:
As = A
As = As[1:]
height = len(As)
for i in range(height):
As[i] = As[i][0:fc] + As[i][fc+1:]
sign = (-1) ** (fc % 2)
sub_det = determHitung(As)
total += sign * A[0][fc] * sub_det
else:
return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
else:
return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
return total
ambil(h)
ambil(y)
pastikan(h)
pastikan(y)
penjumlahan(h)
perkalian(h)
determinan(h)
Ln: 7 Col: 17
Download: --- MB/s
Upload: --- MB/s
23:17


Modul 3 Soal 2.py - C:\Users\asus\Downloads\Modul 3 Soal 2.py (3.8.0)
File Edit Format Run Options Window Help
def buatNol(n,m=None):
if(m==None):
m=n
print("membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
print([[0 for j in range(m)] for i in range(n)])
def buatIdentitas(m):
for i in range(0,m):
for j in range(0,m):
if (i==j):
print('1',end=" ")
else:
print('0',end=" ")
print()
buatNol(4,4)
buatNol(3)
buatIdentitas(4)
Ln: 18 Col: 0
Download: --- MB/s
Upload: --- MB/s
23:18


Modul 3 soal 3.py - C:\Users\asus\Downloads\Modul 3 soal 3.py (3.8.0)
File Edit Format Run Options Window Help
class Node:
def __init__(self, data):
self.data = data
self.next = None
class LinkedList:
def __init__(self):
self.head = None
def tambahDepan(self, new_data):
new_node = Node(new_data)
new_node.next = self.head
self.head = new_node
def tambahAkhir(self, data):
if (self.head == None):
self.head = Node(data)
else:
current = self.head
while (current.next != None):
current = current.next
current.next = Node(data)
return self.head
def tambah(self,data,pos):
node = Node(data)
if not self.head:
self.head = node
elif pos==0:
node.next = self.head
self.head = node
else:
prev = None
current = self.head
current_pos = 0
while(current_pos < pos) and current.next:
prev = current
current = current.next
current_pos +=1
node.next = prev.next
prev.next = node
return self.head
def hapus(self, position):
if self.head == None:
return
temp = self.head
if position == 0:
self.head = temp.next
temp = None
return
for i in range(position):
Ln: 85 Col: 15
Download:
Upload:
23:19


Modul 3 soal 3.py - C:\Users\asus\Downloads\Modul 3 soal 3.py (3.8.0)
File Edit Format Run Options Window Help
def hapus(self, position):
if self.head == None:
return
temp = self.head
if position == 0:
self.head = temp.next
temp = None
return
for i in range(position):
prev = temp
temp = temp.next
if temp is None:
break
if temp is None:
return
if temp.next is None:
return
prev.next = temp.next
temp= None
def cari(self, x):
current = self.head
while current != None:
if current.data == x:
return "True"
current = current.next
return "False"
def display(self):
current = self.head
while current is not None:
print(current.data, end = ' ')
current = current.next
llist = LinkedList()
llist.tambahDepan(21)
llist.tambahDepan(22)
llist.tambahDepan(12)
llist.tambahDepan(14)
llist.tambahDepan(2)
llist.tambahDepan(19)
llist.tambahAkhir(9)
llist.display()
llist.hapus(5)
llist.tambah(1,5)
print(llist.cari(21))
print(llist.cari(29))
llist.display()
Ln: 85 Col: 15
Download:
Upload:
23:19


untitled
File Edit Format Run Options Window Help
class Node:
def __init__(self, data):
self.data = data
self.prev = None
class DoublyLinkedList:
def __init__(self):
self.head = None
def awal(self, new_data):
print("menambah pada awal", new_data)
new_node = Node(new_data)
new_node.next = self.head
if self.head is not None:
self.head.prev = new_node
self.head = new_node
def akhir(self, new_data):
print("menambah pada akhir", new_data)
new_node = Node(new_data)
new_node.next = None
if self.head is None:
new_node.prev = None
self.head = new_node
return
last = self.head
while(last.next is not None):
last = last.next
last.next = new_node
new_node.prev = last
return
def printList(self, node):
print("\nDari Depan :")
while(node is not None):
print(" % d" %(node.data))
last = node
node = node.next
print("\nDari Belakang :")
while(last is not None):
print(" % d" %(last.data))
last = last.prev
llist = DoublyLinkedList()
llist.awal(7)
llist.awal(1)
llist.akhir(6)
llist.akhir(4)
llist.printList(llist.head)
Ln: 13 Col: 34
Download: --- MB/s
Upload: --- MB/s
23:19