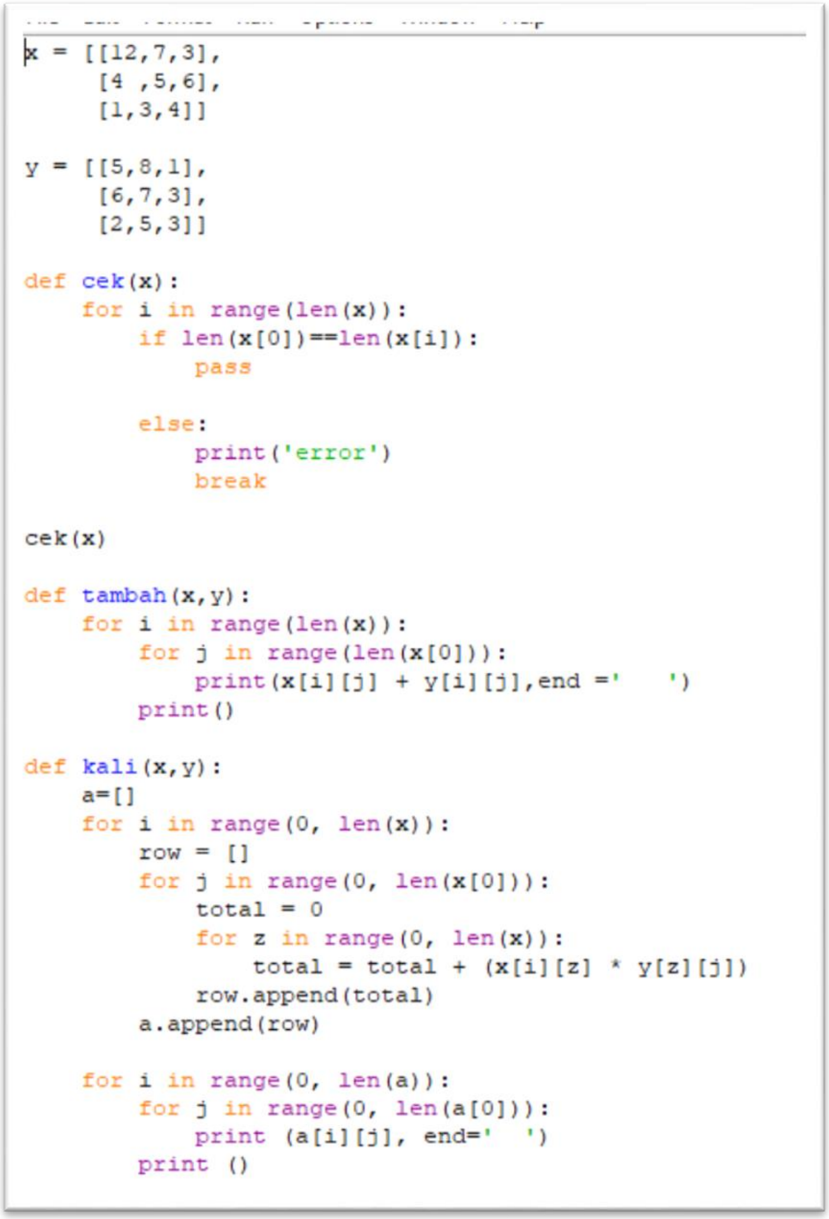
YOGA TRI PRIHATIN

L200170150

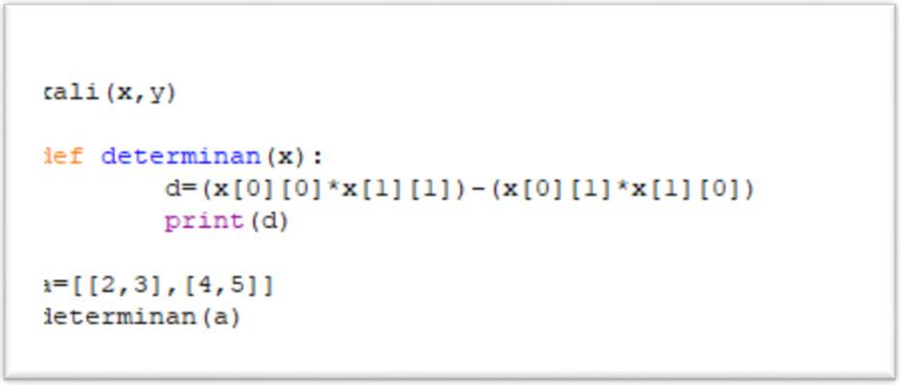
D

MODUL 3

NO1

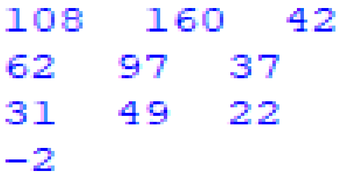


```
x = [[12,7,3],
     [4 ,5,6],
     [1,3,4]]

y = [[5,8,1],
     [6,7,3],
     [2,5,3]]

def cek(x):
    for i in range(len(x)):
        if len(x[0])==len(x[i]):
            pass

        else:
            print('error')
            break

cek(x)

def tambah(x,y):
    for i in range(len(x)):
        for j in range(len(x[0])):
            print(x[i][j] + y[i][j],end ='    ')
        print()

def kali(x,y):
    a=[]
    for i in range(0, len(x)):
        row = []
        for j in range(0, len(x[0])):
            total = 0
            for z in range(0, len(x)):
                total = total + (x[i][z] * y[z][j])
            row.append(total)
        a.append(row)

    for i in range(0, len(a)):
        for j in range(0, len(a[0])):
            print (a[i][j], end='  ')
        print ()
```



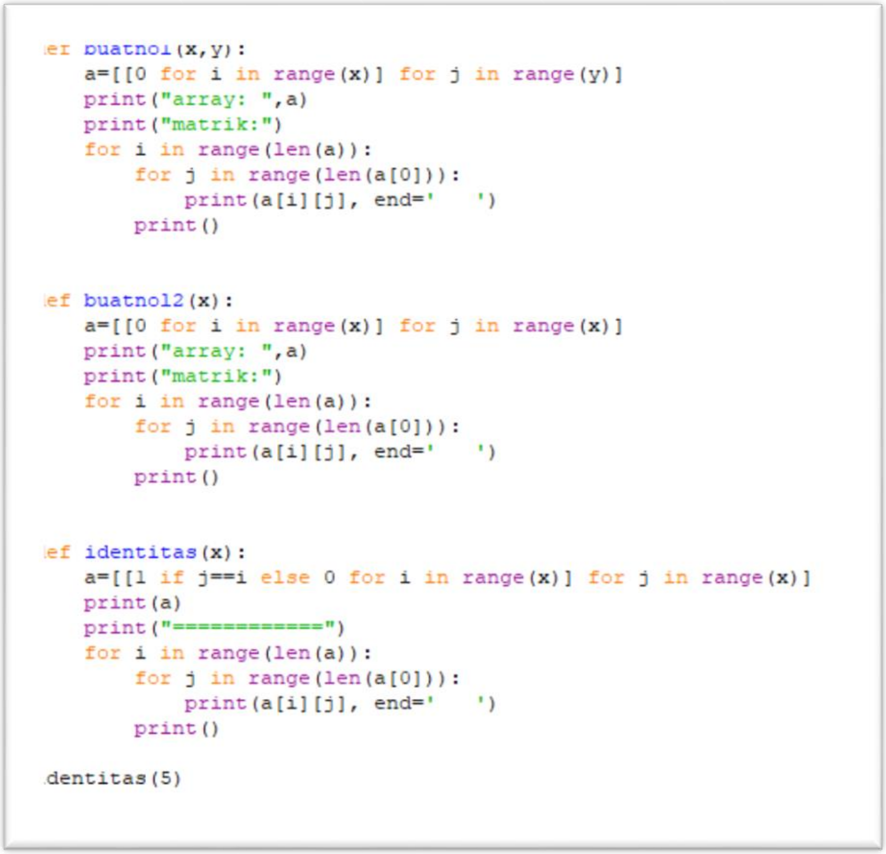
```
kali(x,y)

lef determinan(x):
        d=(x[0][0]*x[1][1])-(x[0][1]*x[1][0])
        print(d)

a=[[2,3],[4,5]]
ieterminan(a)
```

Hasilnya :



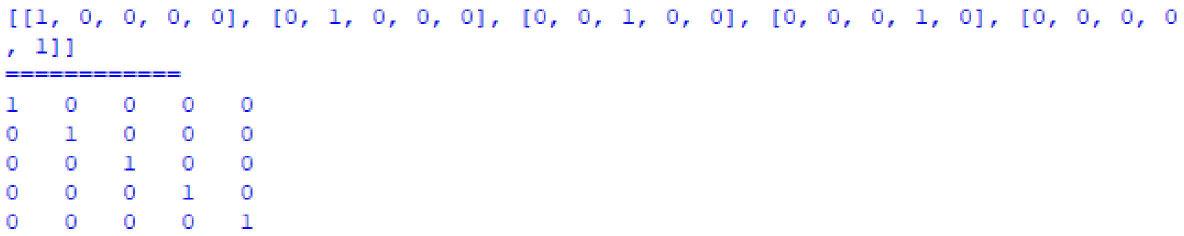
```
108   160   42
62   97   37
31   49   22
-2
```

NO2



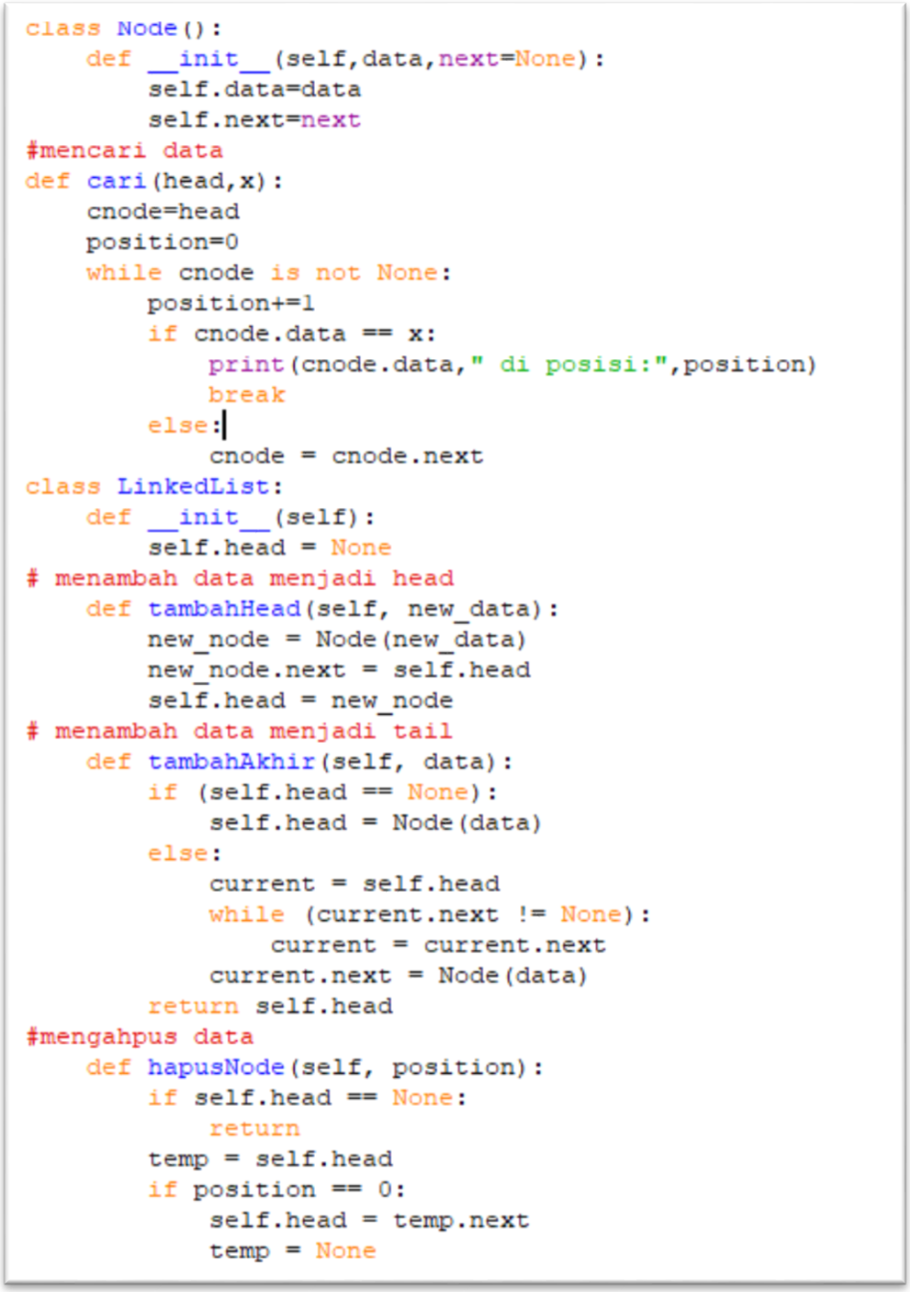
```
ef buatnol(x,y):
    a=[[0 for i in range(x)] for j in range(y)]
    print("array: ",a)
    print("matrik:")
    for i in range(len(a)):
        for j in range(len(a[0])):
            print(a[i][j], end='    ')
        print()


lef buatnol2(x):
    a=[[0 for i in range(x)] for j in range(x)]
    print("array: ",a)
    print("matrik:")
    for i in range(len(a)):
        for j in range(len(a[0])):
            print(a[i][j], end='    ')
        print()


lef identitas(x):
    a=[[1 if j==i else 0 for i in range(x)] for j in range(x)]
    print(a)
    print("============")
    for i in range(len(a)):
        for j in range(len(a[0])):
            print(a[i][j], end='    ')
        print()

dentitas(5)
```

Hasilnya:

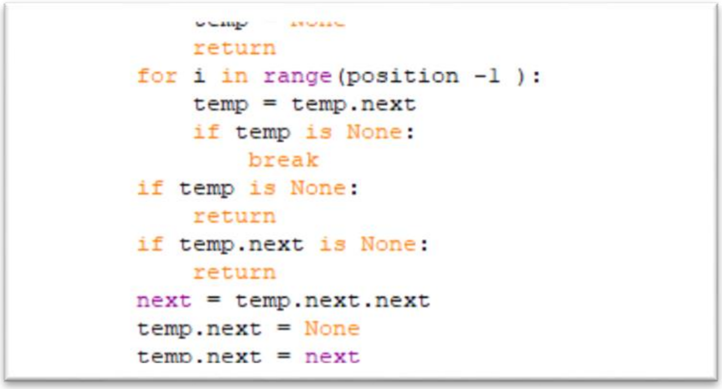


```
[[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 0
, 1]]
============
1    0    0    0    0
0    1    0    0    0
0    0    1    0    0
0    0    0    1    0
0    0    0    0    1
```

NO3

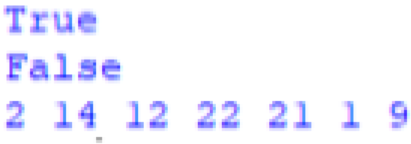


```
class Node():
    def __init__(self,data,next=None):
        self.data=data
        self.next=next
#mencari data
def cari(head,x):
    cnode=head
    position=0
    while cnode is not None:
        position+=1
        if cnode.data == x:
            print(cnode.data," di posisi:",position)
            break
        else:
            cnode = cnode.next
class LinkedList:
    def __init__(self):
        self.head = None
# menambah data menjadi head
    def tambahHead(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
# menambah data menjadi tail
    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
#mengahpus data
    def hapusNode(self, position):
        if self.head == None:
            return
        temp = self.head
        if position == 0:
            self.head = temp.next
            temp = None
```



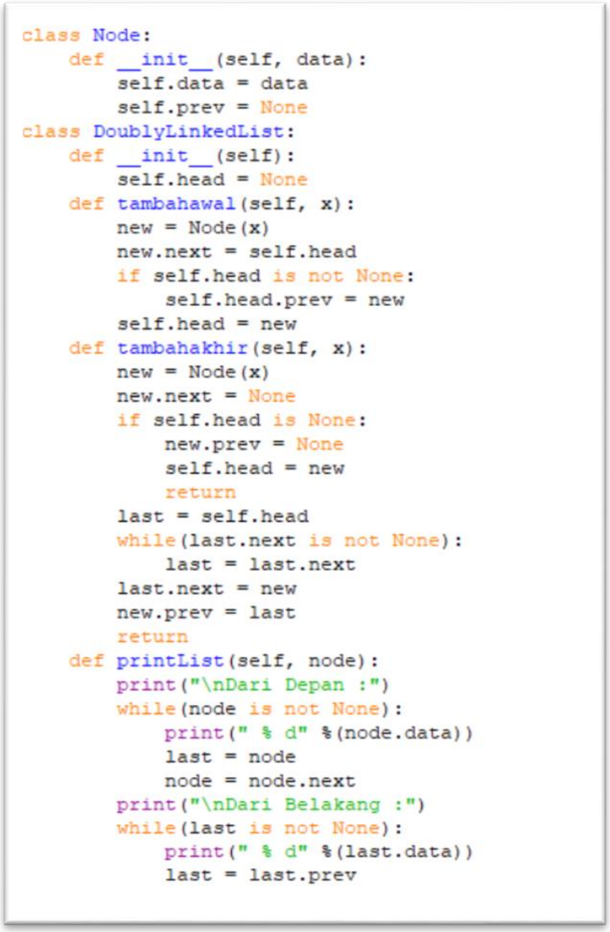
```
            temp = None
            return
        for i in range(position -1 ):
            temp = temp.next
            if temp is None:
                break
        if temp is None:
            return
        if temp.next is None:
            return
        next = temp.next.next
        temp.next = None
        temp.next = next
```

Hasilnya:



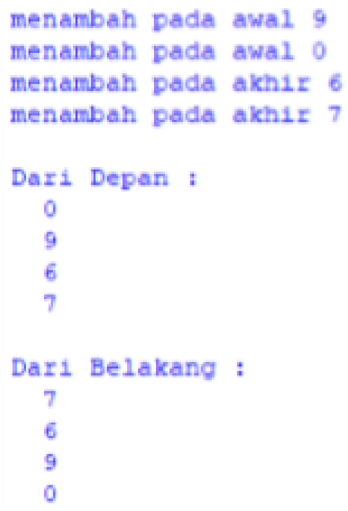
```
True
False
2 14 12 22 21 1 9
```

NO4



```
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
class DoublyLinkedList:
    def __init__(self):
        self.head = None
    def tambahawal(self, x):
        new = Node(x)
        new.next = self.head
        if self.head is not None:
            self.head.prev = new
        self.head = new
    def tambahakhir(self, x):
        new = Node(x)
        new.next = None
        if self.head is None:
            new.prev = None
            self.head = new
            return
        last = self.head
        while(last.next is not None):
            last = last.next
        last.next = new
        new.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
```

Hasilnya:



```
menambah pada awal 9
menambah pada awal 0
menambah pada akhir 6
menambah pada akhir 7

Dari Depan :
  0
  9
  6
  7

Dari Belakang :
  7
  6
  9
  0
```