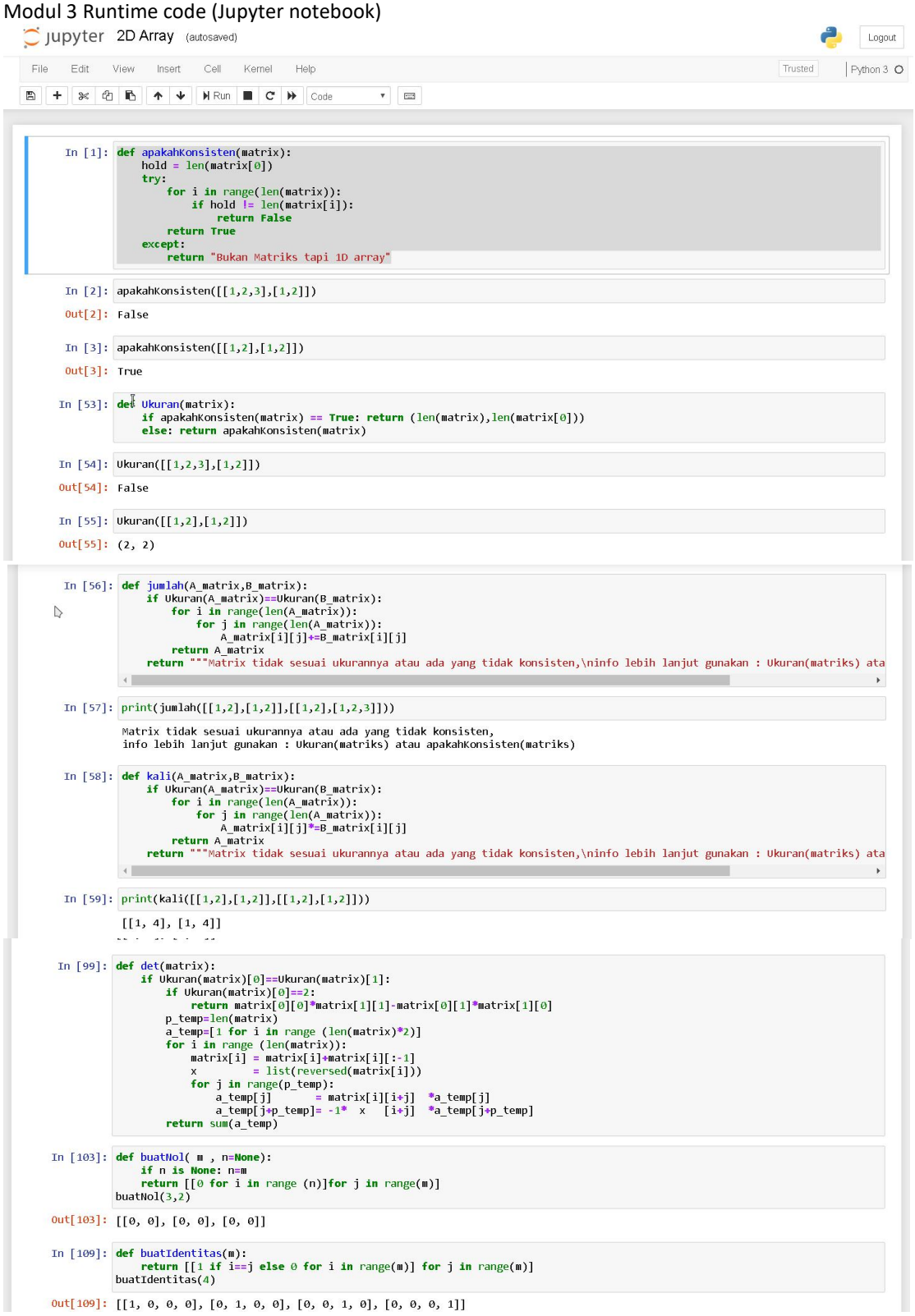
Nama : Herlangga Yusuf Syailendra
NIM : L200180186

Modul 3 Runtime code (Jupyter notebook)



jupyter 2D Array (autosaved) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3

Run Code

In [1]:
```
def apakahKonsisten(matrix):
    hold = len(matrix[0])
    try:
        for i in range(len(matrix)):
            if hold != len(matrix[i]):
                return False
        return True
    except:
        return "Bukan Matriks tapi 1D array"
```

In [2]:
```
apakahKonsisten([[1,2,3],[1,2]])
```
Out[2]: False

In [3]:
```
apakahKonsisten([[1,2],[1,2]])
```
Out[3]: True

In [53]:
```
def Ukuran(matrix):
    if apakahKonsisten(matrix) == True: return (len(matrix),len(matrix[0]))
    else: return apakahKonsisten(matrix)
```

In [54]:
```
Ukuran([[1,2,3],[1,2]])
```
Out[54]: False

In [55]:
```
Ukuran([[1,2],[1,2]])
```
Out[55]: (2, 2)

In [56]:
```
def jumlah(A_matrix,B_matrix):
    if Ukuran(A_matrix)==Ukuran(B_matrix):
        for i in range(len(A_matrix)):
            for j in range(len(A_matrix)):
                A_matrix[i][j]+=B_matrix[i][j]
        return A_matrix
    return """Matrix tidak sesuai ukurannya atau ada yang tidak konsisten,\ninfo lebih lanjut gunakan : Ukuran(matriks) ata
```

In [57]:
```
print(jumlah([[1,2],[1,2]],[[1,2],[1,2,3]]))
```
```
Matrix tidak sesuai ukurannya atau ada yang tidak konsisten,
info lebih lanjut gunakan : Ukuran(matriks) atau apakahKonsisten(matriks)
```

In [58]:
```
def kali(A_matrix,B_matrix):
    if Ukuran(A_matrix)==Ukuran(B_matrix):
        for i in range(len(A_matrix)):
            for j in range(len(A_matrix)):
                A_matrix[i][j]*=B_matrix[i][j]
        return A_matrix
    return """Matrix tidak sesuai ukurannya atau ada yang tidak konsisten,\ninfo lebih lanjut gunakan : Ukuran(matriks) ata
```

In [59]:
```
print(kali([[1,2],[1,2]],[[1,2],[1,2]]))
```
```
[[1, 4], [1, 4]]
```

In [99]:
```
def det(matrix):
    if Ukuran(matrix)[0]==Ukuran(matrix)[1]:
        if Ukuran(matrix)[0]==2:
            return matrix[0][0]*matrix[1][1]-matrix[0][1]*matrix[1][0]
        p_temp=len(matrix)
        a_temp=[1 for i in range (len(matrix)*2)]
        for i in range (len(matrix)):
            matrix[i] = matrix[i]+matrix[i][:-1]
            x         = list(reversed(matrix[i]))
            for j in range(p_temp):
                a_temp[j]       = matrix[i][i+j]  *a_temp[j]
                a_temp[j+p_temp]= -1*  x   [i+j]  *a_temp[j+p_temp]
        return sum(a_temp)
```

In [103]:
```
def buatNol( m , n=None):
    if n is None: n=m
    return [[0 for i in range (n)]for j in range(m)]
buatNol(3,2)
```
Out[103]: [[0, 0], [0, 0], [0, 0]]

In [109]:
```
def buatIdentitas(m):
    return [[1 if i==j else 0 for i in range(m)] for j in range(m)]
buatIdentitas(4)
```
Out[109]: [[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]

In [150]:
```
class Dnode(object):
    def __init__(self,data,next=None):
        self.data=data
        self.next=next
class LList(object):
    def __init__(self,head):
        self.head=head
    def cari(self,yang_dicari):
        pointer=self.head
        counter=0
        while pointer is not None:
            if pointer.data==yang_dicari:
                print("data dengan isi {} ditemukan pada list number {}".format(yang_dicari, counter))
                break
            pointer=pointer.next
            counter+=1
        if pointer==None:
            print("data tidak ditemukan")
    def access(self):
        pointer=self.head
        while pointer is not None:
            print(pointer.data,end=" ")
            pointer=pointer.next
        print("")
    def insert(self,node,position):
        pointer= self.head
        previous= None
        try:
            for i in range(position):
                previous=pointer
                pointer=pointer.next
        except:
            print("Posisi melebihi panjang list")
        if previous is not None:   previous.next=node
        if position==0:            self.head=node
        node.next=pointer
    def tambahDepan(self,node):
        node.next,self.head=self.head,node
    def tambahAkhir(self,node):
        cursor=self.head
        previous=None
        while cursor is not None:
            previous=cursor
            cursor=cursor.next
        previous.next=node
    def hapus(self,position):
        pointer= self.head
        llist=[]
        while pointer is not None:
            llist.append(pointer)
            pointer=pointer.next
        if len(llist)<=position or position<0: print("Posisi salah")
        llist[position].next=None
        llist[position-1].next=llist[position+1]
```

In [151]:
```
a=Dnode(11)
b=Dnode(12)
c=Dnode(13)
e=Dnode(14)
f=Dnode(15)

a.next=b
b.next=c

List=LList(a)
List.access()

List.tambahAkhir(f)
List.insert(e,0)
List.hapus(2)

List.access()
List.cari(11)
```

```
11 12 13
14 11 13 15
data dengan isi 11 ditemukan pada list number 1
```

In [1]:
```
class DNode(object):
    def __init__(self, data, next=None, previous=None):
        self.data    = data
        self.next    = next
        self.previous= previous
```

In [13]:
```
class DoubleLink(object):
    def __init__(self, head):
        self.head    = head
        self.tail    = head
    def tambahDepan(self,node):
        self.head.previous = node
        node.next = self.head
        self.head = node
    def tambahAkhir(self,node):
        prev,cursor = None,self.head
        while cursor is not None:
            prev,cursor = cursor,cursor.next
        prev.next,  node.previous  =  node, prev
        self.tail  =  node
    def kunjungiDepan(self):
        cursor,buffer  =  self.head,[]
        while cursor is not None:
            buffer.append(cursor.data)
            cursor = cursor.next
        print(*buffer, sep=" ")
    def kunjungiBelakang(self):
        cursor,buffer  =  self.tail,[]
        while cursor is not None:
            buffer.append(cursor.data)
            cursor = cursor.previous
        print(*buffer, sep=" ")
```

```
a=DNode(11)
b=DNode(12)
c=DNode(13)
d=DNode(14)

z=DoubleLink(a)

z.tambahAkhir(b)
z.tambahAkhir(c)
z.tambahDepan(d)

z.kunjungiDepan()
z.kunjungiBelakang()
```

```
14 11 12 13
13 12 11 14
```