

SOAL NOMOR 1

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows files in the current workspace, including Soal1.py, Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py, Soal8.py, and Soal1.py.
- Open Editors:** The file Soal1.py is currently open in the editor.
- Code Editor:** The content of Soal1.py is displayed, containing functions for list manipulation:

```
1 #Nama File : Soal1.py
2 #Pembuat : Mohamad Sulkhan Nawawi
3 #Tanggal : 13 November 2023
4 #Deskripsi : fungsi yang menghasilkan nilai maksimum kedua dari sebuah List.
5
6 def Konso(e,L):
7     if L == []:
8         return [e]
9     else :
10        return [e] + L
11
12 def FirstElmt(L) :
13     return L[0]
14
15 def Tail(L):
16     if (L == []):
17         return []
18     else:
19         return L[1:]
20
21 def IsEmpty(L):
22     if (L == []):
23         return True
24     else :
25         return False
26
27 def IsOneElmt(L) :
28     if Tail(L) == []:
29         return True
30
31
```

- List/Dan List Of List:** Shows files List.py, ListOfList.py, Point.py, Soal1.py, Soal2.py, Soal3.py, Soal4.py.
- Debug Console:** A search bar labeled "Filter (e.g. text. i...)" is present.
- Bottom Status Bar:** Displays "Ln 90, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python 3.11.5 64-bit | Go Live | tabnine starter | Prettier".

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows files in the current workspace, including Soal1.py, Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py, Soal8.py, and Soal1.py.
- Editor:** The main editor area displays Python code for list operations. The code includes functions for checking if a list has one element, getting the count of elements, creating a first list, creating a tail list, and checking if an element is an atom.
- Search Bar:** Located at the top center, with the text "List dan List Of List".
- Bottom Status Bar:** Shows file information (Ln 90, Col 1), workspace statistics (Spaces: 4, UTR-B, CRLF), Python version (3.11.5 64-bit), and developer tools (Go Live, tabnine starter).

```
def IsOneElmt(L):
    if Tail(L) == []:
        return True
    else:
        return False

def NbElmt(L):
    if IsEmpty(L):
        return 0
    else:
        return 1 + NbElmt(Tail(L))

def FirstList(S):
    if S == []:
        return None
    else:
        return S[0]

def TailList(S):
    if S == []:
        return None
    else:
        return S[1:]

def IsAtom(S):
    if isinstance(S, (int, float, str, bytes, bool, type(None))):
        return True
    else:
        return False
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows multiple Python files: Soal1.py, Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py, Soal8.py.
- Editor:** The main editor window displays the content of Soal1.py. The code defines several functions: IsAtom, Max, Rember, and MaxList.
- Terminal:** At the bottom, the terminal shows the command PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman\List dan List Of List & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/HP/OneDrive - Universitas Diponegoro/Tugas Kuliah/Semester 1/Dasar Pemrograman/List dan List Of List/Soal1.py".

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the same set of Python files as the first screenshot.
- Editor:** The main editor window now displays the completed code for Soal1.py, including the definition of MaxList2 and its application to two lists.
- Terminal:** The terminal output has been updated to show the results of running the script: MaxList2([2, 6, 10, 99, 100, 20]) and MaxList2([70, 67, 13, 98, 10, 20]).

SOAL NOMOR 2

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files in the current workspace, including Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py, Soal8.py, and Soal1.py.
- Editor:** The main pane displays the following Python code:

```
1 #Nama File : Soal2.py
2 #Pembuat : Mohamad Sulhan Nawawi
3 #Tanggal : 13 November 2023
4 #Deskripsi : sebuah fungsi yang menerima masukan berupa List of integer berdimensi sama,
5 #             lalu menghasilkan List baru yang merupakan hasil perkalian setiap elemen
6 #             di List pertama dan List kedua
7
8 def Konso(e,L):
9     if L == []:
10        return [e]
11    else :
12        return [e] + L
13
14 def FirstElmt(L) :
15     return L[0]
16
17 def Tail(L):
18     return L[1:]
19
20 def IsEmpty(L) :
21     if L == []:
22        return True
23    else :
24        return False
25
26 def NbElmt(L):
27     if IsEmpty(L) :
28        return 0
29    else :
```

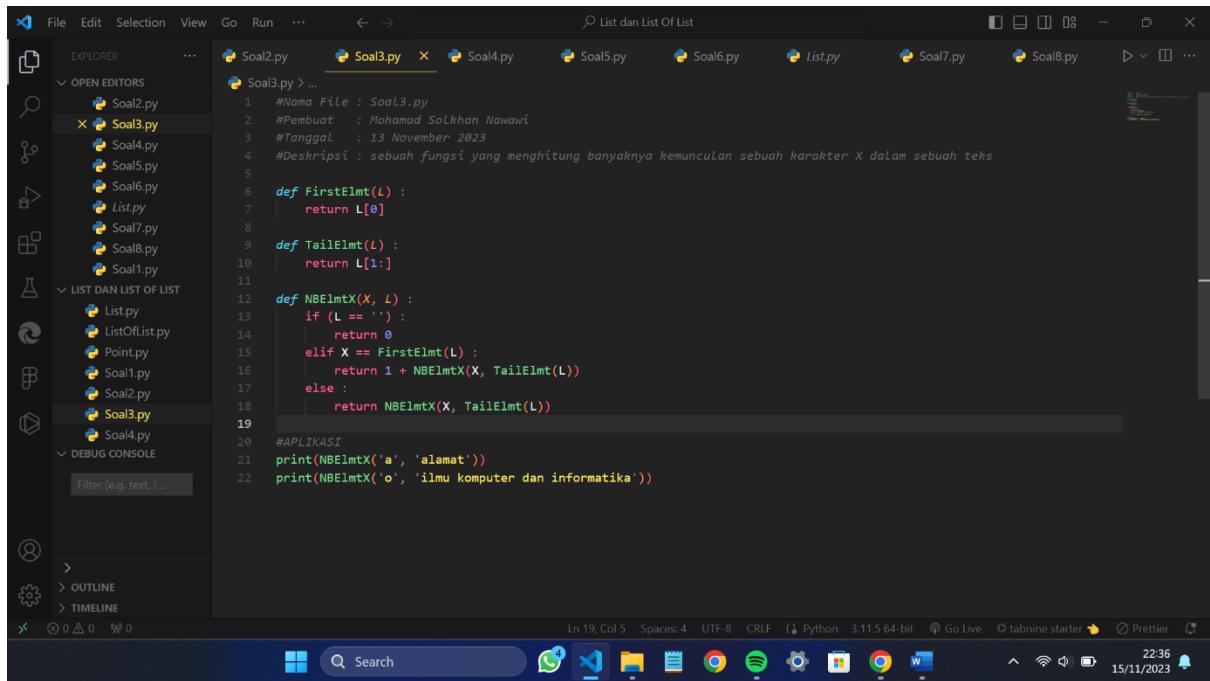
The code defines several functions: Konso, FirstElmt, Tail, IsEmpty, and NbElmt. It also includes a comment describing the task of multiplying two lists element-wise.

The screenshot shows the same code editor interface after the code has been completed. The editor now displays the following code:

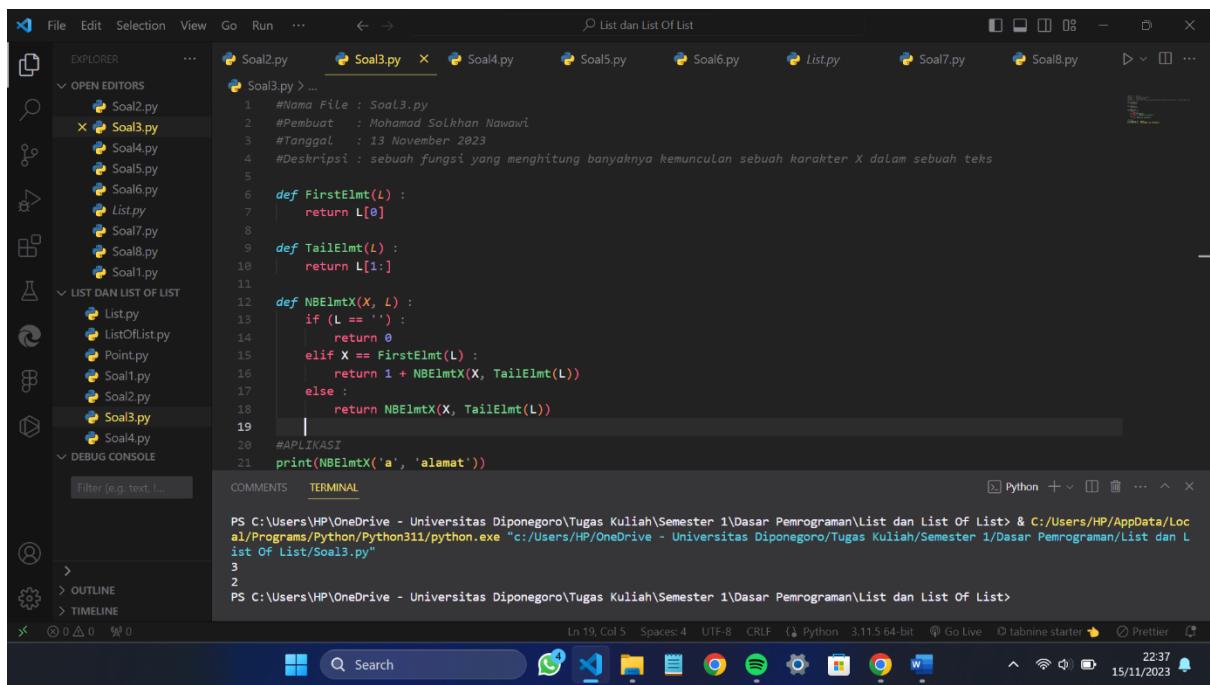
```
25
26 def NbElmt(L):
27     if IsEmpty(L) :
28        return 0
29    else :
30        return 1 + NbElmt(Tail(L))
31
32 def KaliList(L1,L2):
33     if IsEmpty(L1) and IsEmpty(L2):
34        return []
35     elif NbElmt(L1) == NbElmt(L2):
36        return Konso(FirstElmt(L1)*FirstElmt(L2),KaliList(Tail(L1),Tail(L2)))
37     else :
38        return None
39
40 #APLIKASI
41 print(KaliList([2, 4, 6], [1, 2, 3]))
42 print(KaliList([1, 2, 3, 4], [1, 2, 3, 4]))
```

The code now includes a function KaliList that takes two lists as input and returns their element-wise product. The application section at the bottom shows the results of running the code with two test cases.

SOAL NOMOR 3



```
1 #Nama File : Soal3.py
2 #Pembuat : Mohamad Sulhan Nawawi
3 #Tanggal : 13 November 2023
4 #Deskripsi : sebuah fungsi yang menghitung banyaknya kemunculan sebuah karakter X dalam sebuah teks
5
6 def FirstElmt(L) :
7     return L[0]
8
9 def TailElmt(L) :
10    return L[1:]
11
12 def NBEElmtX(X, L) :
13    if (L == '') :
14        return 0
15    elif X == FirstElmt(L) :
16        return 1 + NBEElmtX(X, TailElmt(L))
17    else :
18        return NBEElmtX(X, TailElmt(L))
19
20 #APLIKASI
21 print(NBEElmtX('a', 'alamat'))
22 print(NBEElmtX('o', 'ilmu komputer dan informatika'))
```



```
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman\List dan List Of List> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/HP/OneDrive - Universitas Diponegoro/Tugas Kuliah/Semester 1/Dasar Pemrograman/List dan List Of List/Soal3.py"
3
2
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman\List dan List Of List>
```

SOAL NOMOR 4

VS Code interface showing the Explorer sidebar with files like Soal2.py, Soal3.py, Soal5.py, Soal6.py, List.py, Soal7.py, Soal8.py, and Soal1.py. The current file, Soal4.py, is open in the editor. The code implements functions for list manipulation and palindrome checking.

```
Soal4.py > ...
1 #Nama File : Soal4.py
2 #Pembuat : Mohamad Solkhan Nawawi
3 #Tanggal : 13 November 2023
4 #Deskripsi : fungsi yang mengecek apakah sebuah teks merupakan palindrom
5 #
6
7 def Konso(e,L):
8     if L == []:
9         return [e]
10    else :
11        return [e] + L
12
13 def FirstElmt(L):
14     return L[0]
15
16 def LastElmt(L):
17     return L[-1]
18
19 def Tail(L):
20     return L[1:]
21
22 def Head(L):
23     return L[:-1]
24
25 def IsEmpty(L) :
26     if L == []:
27         return True
28     else :
29         return False
30
31 def Inverse(L):
32     if IsEmpty(L) :
33         return []
34     else :
35         return Konso(LastElmt(L), Inverse(Head(L)))
36
37 def CekPalindrom(L) :
38     if (L == '') :
39         return True
40     else :
41         if FirstElmt(L) == LastElmt(L) :
42             return CekPalindrom(Tail(Head(L)))
43         else :
44             return False
45
46 #APLIKASI
47 print(CekPalindrom('kodok'))
48 print(CekPalindrom('daspro'))
```

VS Code interface showing the Explorer sidebar with files like Soal2.py, Soal3.py, Soal5.py, Soal6.py, List.py, Soal7.py, Soal8.py, and Soal1.py. The current file, Soal4.py, is open in the editor. The code has been modified to include additional functions and logic for palindrome checking.

```
Soal4.py > ...
25 def IsEmpty(L) :
26     if L == []:
27         return True
28     else :
29         return False
30
31 def Inverse(L):
32     if IsEmpty(L) :
33         return []
34     else :
35         return Konso(LastElmt(L), Inverse(Head(L)))
36
37 def CekPalindrom(L) :
38     if (L == '') :
39         return True
40     else :
41         if FirstElmt(L) == LastElmt(L) :
42             return CekPalindrom(Tail(Head(L)))
43         else :
44             return False
45
46 #APLIKASI
47 print(CekPalindrom('kodok'))
48 print(CekPalindrom('daspro'))
```

File Edit Selection View Go Run ...

EXPLORER

- OPEN EDITORS
- Soal2.py
- Soal3.py
- Soal4.py
- Soal5.py
- Soal6.py
- List.py
- Soal7.py
- Soal8.py
- Soal1.py

LIST DAN LIST OF LIST

- List.py
- ListOfList.py
- Point.py
- Soal1.py
- Soal2.py
- Soal3.py
- Soal4.py

DEBUG CONSOLE

Filter (e.g. text, !...)

COMMENTS TERMINAL

```
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman\List dan List Of List & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/HP/OneDrive - Universitas Diponegoro/Tugas Kuliah/Semester 1/Dasar Pemrograman/List dan List Of List/Soal4.py"
True
False
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman\List dan List Of List>
```

Ln 5, Col 87 Spaces: 2 UTRF-8 CRLF ⚡ Python 3.11.5 64-bit ⚡ Go Live ⚡ tabnine starter ⚡ Prettier ⚡

SOAL NOMOR 5

```
#Nama File : Soal5.py
#Pembuat : Mohamad Solikhun Nawawi
#Tanggal : 13 November 2023
#Deskripsi : sebuah List of point (menggunakan tipe bentukan point),
#           lalu buatlah sebuah fungsi yang menerima masukan sebuah point P
#           dan List of point LP, dan mengembalikan sebuah point di List of point LP
#           yang memiliki jarak Euclidean terdekat dengan poin P.
#           Jika ada Lebih dari satu point yang terdekat, cukup kembalikan salah satunya saja.

def Point(a,b) :
    return [a,b]

def Absis(P) :
    return P[0]

def Ordinat(P) :
    return P[1]

def Jarak(P1,P2) :
    return ((Absis(P1)-Absis(P2))**2 + (Ordinat(P1)-Ordinat(P2))**2)**0.5

def FirstElmt(L) :
    if L == [] :
        return None
    else :
        return L[0]

def Tail(L):
    return L[1:]
```

File Edit Selection View Go Run ... ← → ⌂ List dan List Of List

EXPLORER OPEN EDITORS Soal2.py Soal3.py Soal4.py Soal5.py Soal6.py List.py Soal7.py Soal8.py Soal1.py

Soal5.py > ...

```
28     def Tail(L):
29         return L[1:]
30
31     def IsOneElmt(L):
32         return (Tail(L) == [])
33
34     #REALISASI
35     def NearestPoint(P, LP):
36         if IsOneElmt(LP):
37             return FirstElmt(LP)
38         else:
39             if Jarak(P, NearestPoint(P, Tail(LP))) < Jarak(P, FirstElmt(LP)):
40                 return NearestPoint(P, Tail(LP))
41             else:
42                 return FirstElmt(LP)
43
44     #APLIKASI
45     print(NearestPoint(Point(1,2),[Point(0,0),Point(3,4),Point(6,7),Point(1,1),Point(2,1)]))
```

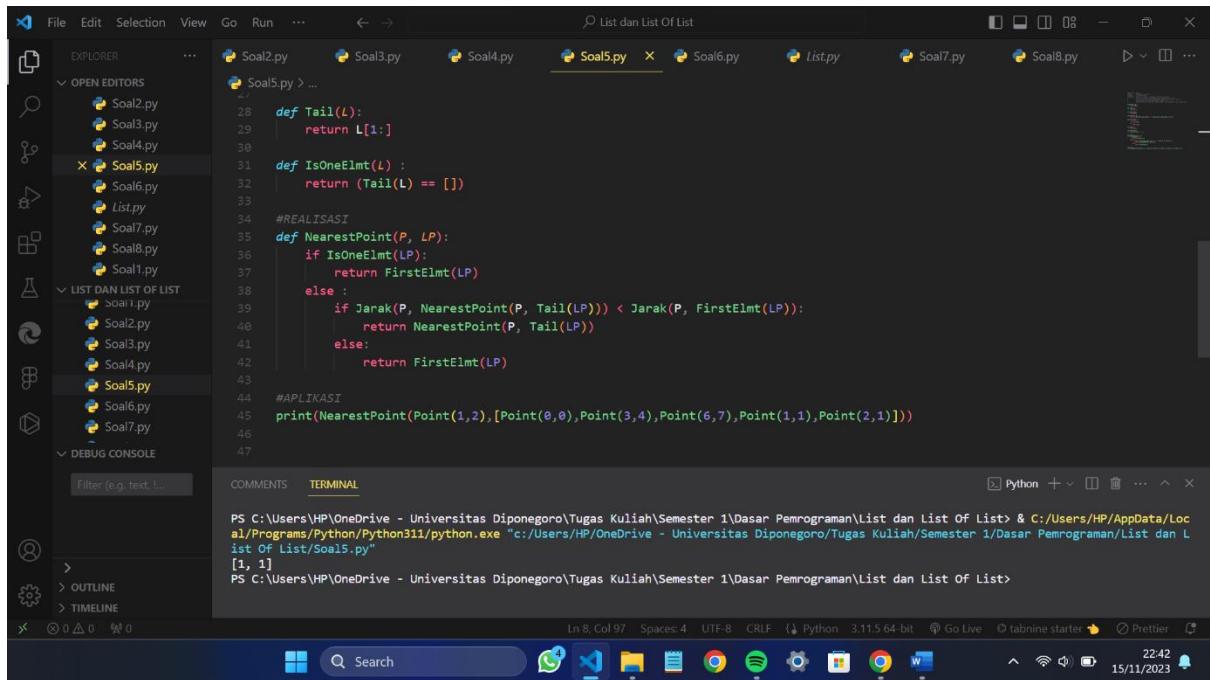
COMMENTS TERMINAL

```
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman>List dan List Of List & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/HP/OneDrive - Universitas Diponegoro/Tugas Kuliah/Semester 1/Dasar Pemrograman/List dan List Of List/Soal5.py"
[1, 1]
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman>List dan List Of List>
```

Ln 8, Col 97 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live tabnine starter Prettier 22:42 15/11/2023

Filter (e.g. text...) Python + ⌂

OUTLINE TIMELINE



SOAL NOMOR 6

File Edit Selection View Go Run ... ← → ⌂ List dan List Of List

EXPLORER OPEN EDITORS Soal2.py Soal3.py Soal4.py Soal5.py Soal6.py List.py Soal7.py Soal8.py Soal1.py

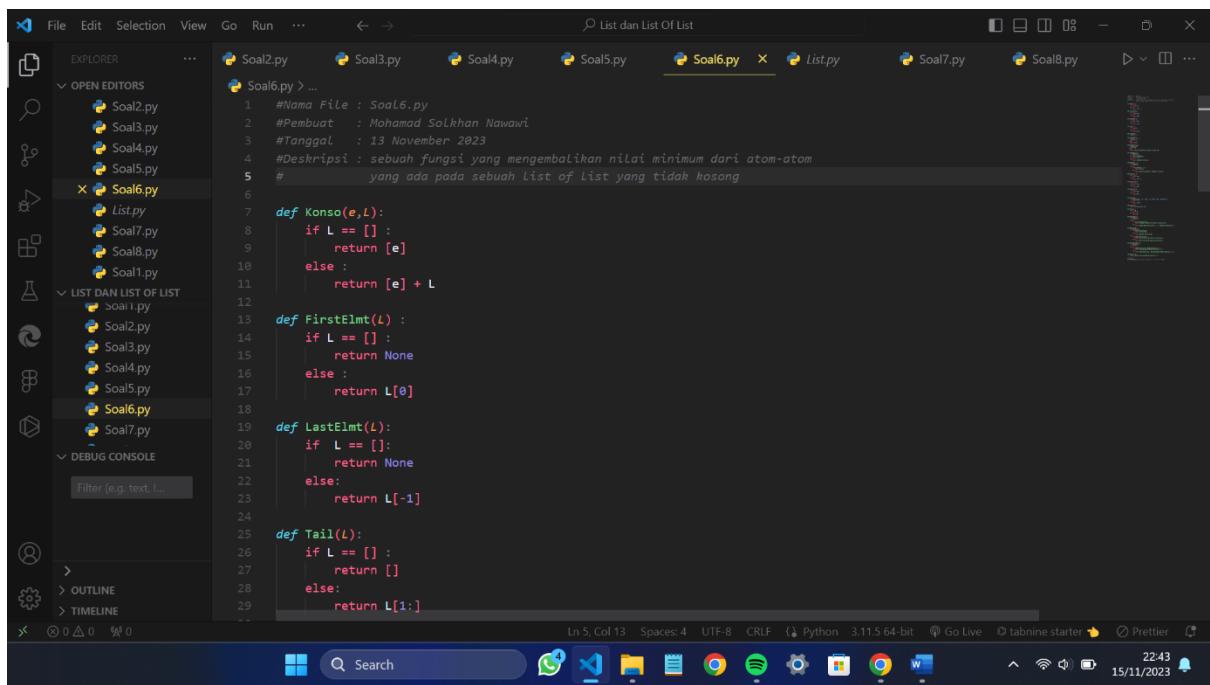
Soal6.py > ...

```
1     #Nama File : Soal6.py
2     #Pembuat : Mahamad Sulhan Nawawi
3     #Tanggal : 13 November 2023
4     #Deskripsi : sebuah fungsi yang mengembalikan nilai minimum dari atom-atom
5     #              yang ada pada sebuah list of list yang tidak kosong
6
7     def Konso(e,L):
8         if L == []:
9             return [e]
10        else:
11            return [e] + L
12
13     def FirstElmt(L):
14         if L == []:
15             return None
16         else:
17             return L[0]
18
19     def LastElmt(L):
20         if L == []:
21             return None
22         else:
23             return L[-1]
24
25     def Tail(L):
26         if L == []:
27             return []
28         else:
29             return L[1:]
```

Ln 5, Col 13 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live tabnine starter Prettier 22:43 15/11/2023

Filter (e.g. text...) Python + ⌂

OUTLINE TIMELINE



VS Code interface showing Python code for list operations. The code defines several functions:

```
def IsEmpty(L):
    return L == []

def IsOneElmt(L):
    if Tail(L) == []:
        return True
    else:
        return False

def Konkat(L1,L2):
    if IsEmpty(L1):
        return L2
    else:
        return Konso(FirstElmt(L1),Konkat(Tail(L1),L2))

def IsMember(X,L):
    if IsEmpty(L):
        return False
    elif X == FirstElmt(L):
        return True
    else:
        return IsMember(X,Tail(L))

def Rember(e, L):
    if IsEmpty(L):
        return []
    else:
        if FirstElmt(L) == e:
            return Tail(L)
        else:
            return Konso(FirstElmt(L), Rember(e, Tail(L)))
```

The code is located in the file `Soal6.py` in the `OPEN EDITORS` panel. The status bar at the bottom shows the file is 37 lines long, in column 9, with 4 spaces, using UTF-8 encoding, and is a 3.11.5 64-bit Python tab.

VS Code interface showing continuation of Python code for list operations. The code defines more functions:

```
def Rember(e, L):
    if IsEmpty(L):
        return []
    else:
        if FirstElmt(L) == e:
            return Tail(L)
        else:
            return Konso(FirstElmt(L), Rember(e, Tail(L)))

def KonsoLo(L,S):
    if L == []:
        return [S]
    else:
        return [S] + L

def FirstList(S):
    if S == []:
        return None
    else:
        return S[0]

def TailList(S):
    if S == []:
        return None
    else:
        return S[1:]

def IsAtom(S) :
```

The code is located in the file `Soal6.py` in the `OPEN EDITORS` panel. The status bar at the bottom shows the file is 37 lines long, in column 9, with 4 spaces, using UTF-8 encoding, and is a 3.11.5 64-bit Python tab.

The screenshot shows a code editor interface with multiple tabs open. The active tab is 'Soal6.py' which contains the following code:

```
def IsOneElmt(S):
    if isinstance(S, (int, float, str, bytes, bool, type(None))):
        return True
    else:
        return False

def IsList(S):
    return isinstance(S, list)

def Max(a,b):
    if a >= b :
        return a
    else:
        return b

def KonkatAll(L):
    if IsEmpty(L):
        return []
    else:
        if IsList(FirstList(L)):
            return KonkatAll(Konkat(FirstList(L), TailList(L)))
        else:
            return Konkat(Konso(FirstList(L), []), KonkatAll(TailList(L)))

def MaxList(S):
    if IsOneElmt(S):
        if IsAtom(FirstList(S)):
            return FirstList(S)
        else:
            return MaxList(FirstList(S))
    else:
        if IsAtom(FirstList(S)):
            return Max(FirstList(S),MaxList(TailList(S)))
        else:
            return Max(FirstList(S),MaxList(TailList(S)))

def DescendingUnik(L):
    if IsEmpty(L) :
        return []
    else:
        if IsMember(MaxList(L),Rember(MaxList(L),L)):
            return DescendingUnik(Rember(MaxList(L), L))
        else:
            return Konso(MaxList(L), DescendingUnik(Rember(MaxList(L), L)))

def Minlist(L):
    return LastElmt(DescendingUnik(KonkatAll(L)))

#APLIKASI
print(Minlist([3, [2, 4, 5], [6, 3], [6, 4, 1, 2], 7, [2]]))
```

The code implements various functions for list manipulation, including checking for single elements, lists, and atoms; finding maximum values; concatenating lists; and finding unique elements in a descending order.

This screenshot shows the continuation of the code from the previous editor. The active tab is still 'Soal6.py' and the code has been updated to include the 'DescendingUnik' and 'Minlist' functions, along with an application example at the bottom.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "OPEN EDITORS" section: Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py (highlighted), Soal7.py, Soal8.py, and Soal1.py.
- Editor:** Displays Python code for "Soal6.py".

```
119     if IsEmpty(L) :
120         return []
121     else:
122         if IsMember(MaxList(L), Rember(MaxList(L), L)):
123             return DescendingUnik(Rember(MaxList(L), L))
124         else:
125             return Konso(MaxList(L), DescendingUnik(Rember(MaxList(L), L)))
126
127     def MinList(L):
128         return LastElmt(DescendingUnik(KonkatAll(L)))
129
130 #APLIKASI
131 print(MinList([3, [2, 4, 5], [6, 3], [6, 4, 1, 2], 7, [2]]))
```
- Terminal:** Shows the command line output for "Soal6.py".

```
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman>List dan List Of List & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/HP/OneDrive - Universitas Diponegoro/Tugas Kuliah/Semester 1/Dasar Pemrograman/List dan List Of List/Soal6.py"
1
PS C:\Users\HP\OneDrive - Universitas Diponegoro\Tugas Kuliah\Semester 1\Dasar Pemrograman>List dan List Of List>
```
- Bottom Bar:** Includes icons for file operations, search, and system status.

SOAL NOMOR 7

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "OPEN EDITORS" section: Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, Soal7.py (highlighted), Soal8.py, and Soal1.py.
- Editor:** Displays Python code for "Soal7.py".

```
1 #Nama File : Soal7.py
2 #Pembuat : Mohamad Sulhan Nawawi
3 #Tanggal : 13 November 2023
4 #Deskripsi : sebuah fungsi yang mengembalikan banyaknya elemen atom bernilai ganjil
5 #          yang ada pada sebuah List of List. Jika List kosong, maka akan mengembalikan nilai 0.
6
7 def Konso(e,L):
8     if L == []:
9         return [e]
10    else :
11        return [e] + L
12
13 def FirstElmt(L):
14     if L == []:
15         return None
16     else :
17         return L[0]
18
19 def Tail(L):
20     if L == []:
21         return []
22     else:
23         return L[1:]
24
25 def Konkat(L1,L2) :
26     if IsEmpty(L1) :
27         return L2
28     else :
29         return Konso(FirstElmt(L1),Konkat(Tail(L1),L2))
```
- Bottom Bar:** Includes icons for file operations, search, and system status.

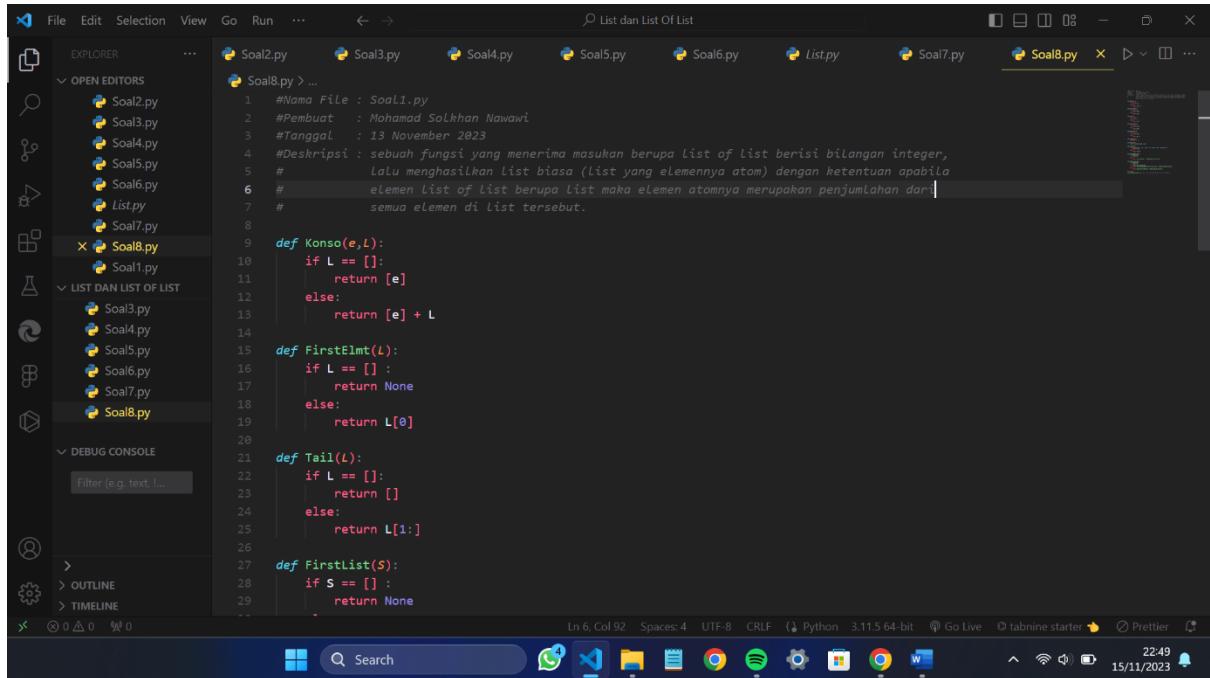
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "OPEN EDITORS" section: Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py (marked with an X), Soal8.py, Soal1.py.
- Editor:** The main editor window displays Python code for list operations. The code includes functions like FirstList, TailList, IsEmpty, IsAtom, IsList, KonkatAll, and NBODds.
- Terminal:** The terminal at the bottom shows the command "python Soal7.py" being run.
- Status Bar:** Shows "Ln 5, Col 99" and "Python 3.11.5 64-bit".

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "OPEN EDITORS" section: Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py (marked with an X), Soal8.py, Soal1.py.
- Editor:** The main editor window displays the same Python code as the first screenshot.
- Terminal:** The terminal at the bottom shows the output of running the code: "NBODds([3, [2, 4, 5], [6, 3], [6, 4, 1, 2], 7, [2]])" and "NBODds([3, [2, 4, 5], [6, 3], [6, 4, 1, 2], 7, [21]])".
- Status Bar:** Shows "Ln 5, Col 99" and "Python 3.11.5 64-bit".

SOAL MOMOR 8



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows multiple Python files: Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py, and Soal8.py.
- Editor:** The current file is Soal8.py, which contains the following code:

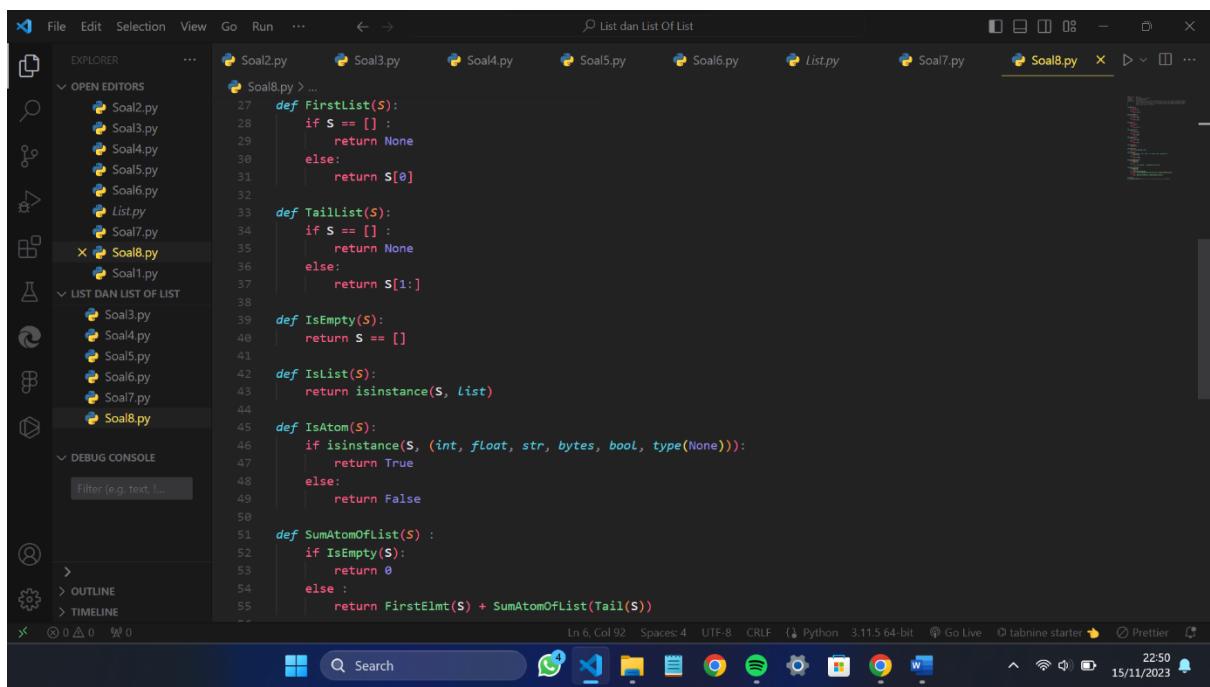
```
#Nama File : Soal1.py
#Rembuk : Mohamad Sulkhan Nawawi
#Tanggal : 13 November 2023
#Deskripsi : sebuah fungsi yang menerima masukan berupa List of list berisi bilangan integer,
#             lalu menghasilkan List biasa (list yang elemennya atom) dengan ketentuan apabila
#             elemen List of List berupa List maka elemen atomnya merupakan penjumlahan dari
#             semua elemen di List tersebut.

def Konso(e,L):
    if L == []:
        return [e]
    else:
        return [e] + L

def FirstElmt(L):
    if L == []:
        return None
    else:
        return L[0]

def Tail(L):
    if L == []:
        return []
    else:
        return L[1:]

def FirstList(S):
    if S == []:
        return None
    else:
```



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows multiple Python files: Soal2.py, Soal3.py, Soal4.py, Soal5.py, Soal6.py, List.py, Soal7.py, and Soal8.py.
- Editor:** The current file is Soal8.py, which contains the following code:

```
def Firstlist(S):
    if S == []:
        return None
    else:
        return S[0]

def Taillist(S):
    if S == []:
        return None
    else:
        return S[1:]

def IsEmpty(S):
    return S == []

def IsList(S):
    return isinstance(S, list)

def IsAtom(S):
    if isinstance(S, (int, float, str, bytes, bool, type(None))):
        return True
    else:
        return False

def SumAtomOfList(S) :
    if IsEmpty(S):
        return 0
    else :
        return FirstElmt(S) + SumAtomOfList(Tail(S))
```

The screenshot shows a Windows desktop environment. In the foreground, a code editor window for Python is open, displaying a script named `Soal8.py`. The code implements a recursive function `MakeListAtom` to flatten lists. The terminal tab in the editor shows the command `python Soal8.py` being run, and the output is a list of integers: [3, 11, 4, 13, 7, 2]. The taskbar at the bottom contains icons for File Explorer, Edge browser, FileZilla, Spotify, Task View, and others. The system tray shows the date as 15/11/2023 and the time as 22:50.

```
def MakeListAtom(S) :
    if IsEmpty(S):
        return []
    elif IsList(FirstElmt(S)):
        return Konso(SumAtomOfList(FirstList(S)), MakeListAtom(Tail(S)))
    else:
        return Konso(FirstElmt(S), MakeListAtom(Tail(S)))

#APLIKASI
print(MakeListAtom(([3, [2, 4, 5], [1, 3], [6, 4, 1, 2], 7, [2]])))
```