Laporan

Struktur Data

Binary Search Tree

Institut Teknologi Kalimantan

Dosen Pengampu:

Muchammad Chandra Cahyo Utomo, M.Kom. 199205202019031013

Disusun Oleh :

Guntur Wisnu Saputra	11211042
Muhammad Insan Kamil	11211058
Muhammad Ricky Zakaria	11211062
Ramadhan Djibran Sanjaya	11211070
Rangga Hermawan	11211071
Rendy Pernanda	11211074

24 Oktober 2022

# Source Code

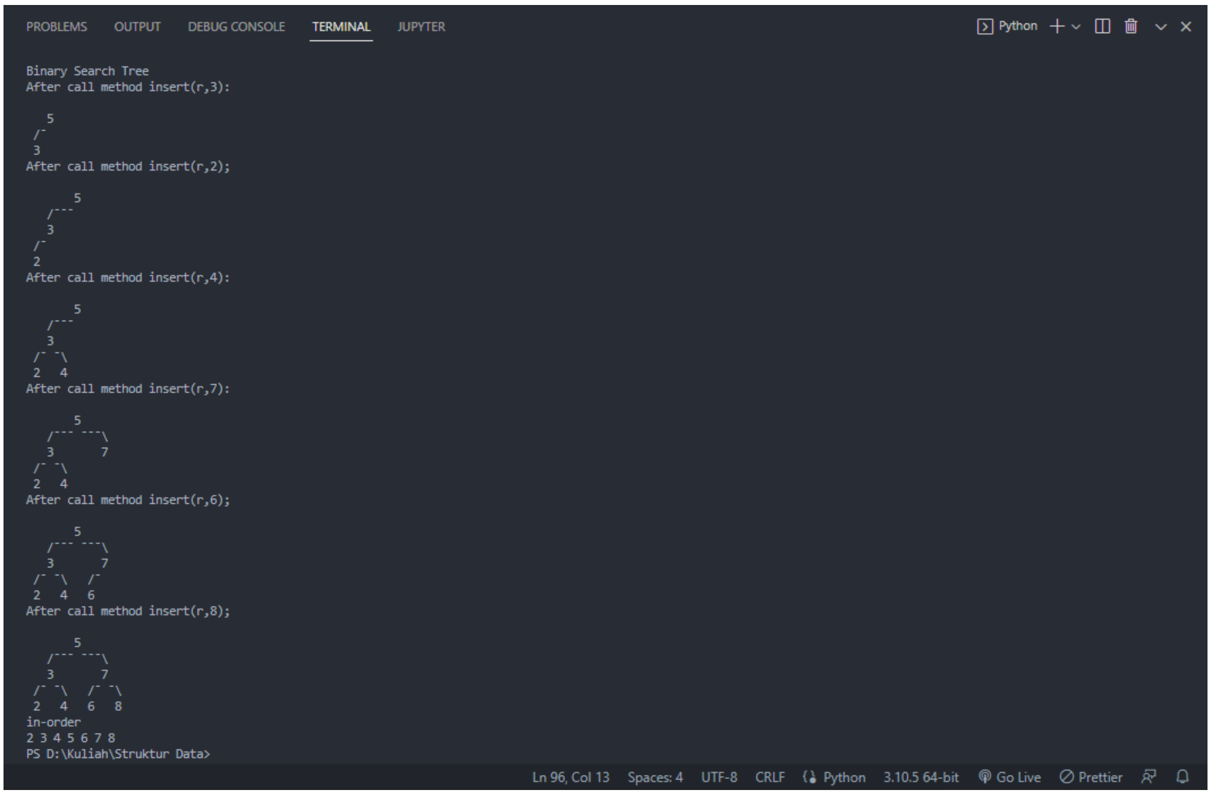
No. BinarySearchTree.py

```
class Node:
    def __init__(self, value):
        self.__left = None
        self.__right = None
        self.__value = value
    def setLeft(self,left):
        self.__left = left
    def setRight(self,right):
        self.__right = right
    def setValue(self, value):
        self.__value = value
    def getLeft(self):
        return self.__left
    def getRight(self):
        return self.__right
    def getValue(self):
        return self.__value

def insert(root, value):
    if root is None:
        return Node(value)
    else:
        if root.getValue() == value:
            return root
        elif root.getValue() < value:
            root.setRight(insert(root.getRight(), value))
        else:
            root.setLeft(insert(root.getLeft(), value))
    return root

def PrintTree(root):
    def height(root):
        return 1 + max(height(root.getLeft()), height(root.getRight())) if root else -1
    nlevels = height(root)
    width =  pow(2,nlevels+1)

    q=[(root,0,width,'c')]
    levels=[]

    while(q):
        node,level,x,align= q.pop(0)
        if node:
            if len(levels)<=level:
                levels.append([])

            levels[level].append([node,level,x,align])
            seg= width//(pow(2,level+1))
            q.append((node.getLeft(),level+1,x-seg,'l'))
            q.append((node.getRight(),level+1,x+seg,'r'))

    for i,l in enumerate(levels):
        pre=0
        preline=0
        linestr=''
        pstr=''
        seg= width//(pow(2,i+1))
        for n in l:
            valstr= str(n[0].getValue())
            if n[3]=='r':
                linestr+=' '*(n[2]-preline-1-seg-seg//2)+ '¯'*(seg +seg//2)+'\\'
                preline = n[2]
            if n[3]=='l':
               linestr+=' '*(n[2]-preline-1)+'/' + '¯'*(seg+seg//2)
               preline = n[2] + seg + seg//2
            pstr+=' '*(n[2]-pre-len(valstr))+valstr
            pre = n[2]
        print(linestr)
        print(pstr)

def inorder(root):
    if root:
        inorder(root.getLeft())
        print(root.getValue(), end=" ")
        inorder(root.getRight())

r = Node(5)
print()
print("Binary Search Tree")
print("After call method insert(r,3):")
r = insert(r, 3)
PrintTree(r)
print("After call method insert(r,2);")
r = insert(r, 2)
PrintTree(r)
print("After call method insert(r,4):")
r = insert(r, 4)
PrintTree(r)
print("After call method insert(r,7):")
r = insert(r, 7)
```

```
PrintTree(r)
print("After call method insert(r,6);")
r = insert(r, 6)
PrintTree(r)
print("After call method insert(r,8);")
r = insert(r, 8)
PrintTree(r)
print("in-order")
inorder(r)
```

#Hasil Run

BinarySearchTree.py



```
Binary Search Tree
After call method insert(r,3):

   5
 /¯
 3
After call method insert(r,2);

       5
   /¯¯¯
   3
 /¯
 2
After call method insert(r,4):

       5
   /¯¯¯
   3
 /¯ ¯\
 2   4
After call method insert(r,7):

       5
   /¯¯¯ ¯¯¯\
   3       7
 /¯ ¯\
 2   4
After call method insert(r,6);

       5
   /¯¯¯ ¯¯¯\
   3       7
 /¯ ¯\   /¯
 2   4   6
After call method insert(r,8);

       5
   /¯¯¯ ¯¯¯\
   3       7
 /¯ ¯\   /¯ ¯\
 2   4   6   8
in-order
2 3 4 5 6 7 8
PS D:\Kuliah\Struktur Data>
```