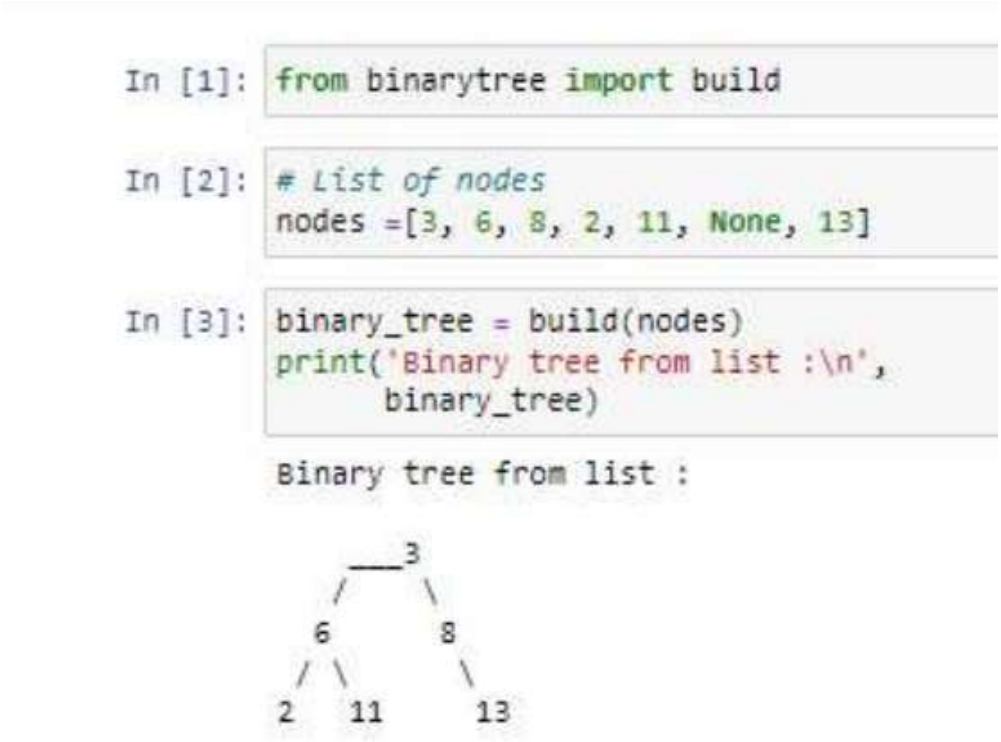
MUHAMMAD TARMIDZI BARIQ

51422161

3IA11

M5



from binarytree import build\

Pada baris ini, pustaka binarytree diimpor, dan fungsi build digunakan untuk membuat pohon biner dari sebuah list.

nodes = [3, 6, 8, 2, 11, None, 13]

Ini adalah daftar nilai-nilai node yang akan digunakan untuk membangun pohon biner. Nilai None digunakan untuk menandakan bahwa posisi itu tidak memiliki node (atau dengan kata lain, kosong).

binary_tree = build(nodes)

Fungsi build(nodes) digunakan untuk membangun pohon biner dari daftar yang telah ditentukan. Pohon ini disusun secara otomatis berdasarkan urutan daftar yang diberikan.

print('Binary tree from list :\n', binary_tree)

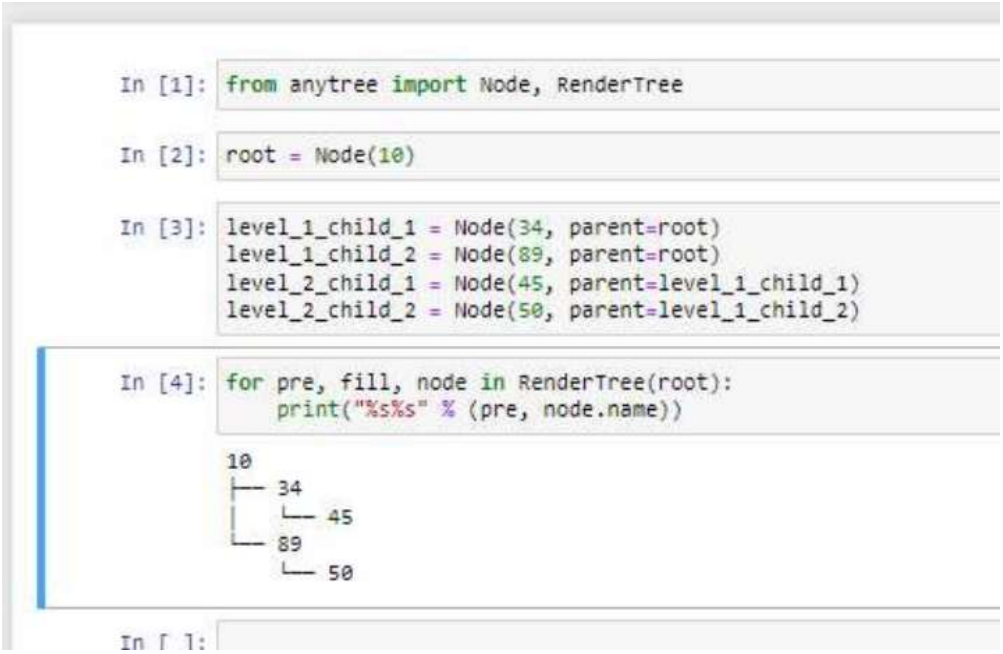
Baris ini mencetak pohon biner yang sudah dibangun ke layar. Hasil dari binary_tree adalah representasi visual dari struktur pohon tersebut.

```
In [4]: # Getting list of nodes from
        # binarytree
        print('\nList from binary tree :',
              binary_tree.values)

        List from binary tree : [3, 6, 8, 2, 11, None, 13]
```

print('\nList from binary tree :', binary_tree.values)

Pada baris ini, kita menggunakan properti values dari objek binary_tree untuk mengambil daftar nilai-nilai yang ada di pohon biner yang sudah dibangun sebelumnya. Ini menghasilkan daftar nilai yang membentuk pohon tersebut.



from anytree import Node, RenderTree

Di sini, kita mengimpor Node dan RenderTree dari pustaka anytree. Node digunakan untuk membuat simpul pada pohon, dan RenderTree digunakan untuk menggambar (menampilkan) pohon dalam format yang mudah dibaca.

root = Node(10)

Baris ini mendefinisikan simpul root dengan nilai 10.

level_1_child_1 = Node(34, parent=root)

level_1_child_2 = Node(89, parent=root)

level_2_child_1 = Node(45, parent=level_1_child_1)

level_2_child_2 = Node(50, parent=level_1_child_2)

Pada baris ini, kita membuat beberapa simpul anak. level_1_child_1 dan level_1_child_2 adalah anak langsung dari root, sedangkan level_2_child_1 adalah anak dari level_1_child_1, dan level_2_child_2 adalah anak dari level_1_child_2.

```
for pre, fill, node in RenderTree(root):
    print("%s%s" % (pre, node.name))
```

Bagian ini menggunakan RenderTree untuk menampilkan pohon yang dimulai dari root. Fungsi RenderTree(root) menghasilkan tiga nilai untuk setiap simpul: pre (penanda indentasi), fill (penanda penghubung antara simpul), dan node (simpul itu sendiri). node.name berisi nilai dari simpul tersebut. Dalam hal ini, setiap simpul akan dicetak dengan indentasi yang menunjukkan posisinya dalam struktur pohon.