Nama : Naufal Luthfi Nurhamid

NIM : 1900018282

Slot : Selasa 10.30

Kode soal : C

1. Ubah kode

Source Code setelah diubah :

```
from ai_pkg.search import *

start = 'Frankfurt'
goal = 'Numberg'

city_map = Graph(dict(
    Frankfurt = dict( Mannheim=85, Wurzburg=217, Kassel=173),
    Mannheim = dict(Karisruhe=80),
    Wurzburg= dict(Erfurt=186, Numberg=103),
    Kassel= dict(Munchen=502),
    Karisruhe = dict(Augsburg=250),
    Numberg = dict(Munchen=167, Stuttgart=183),
    Augsburg = dict(Munchen=84),
    ), directed=True)

class CityProblem(Problem):
    def __init__(self, initial, goal, graph):
        Problem.__init__(self, initial, goal)
        self.graph = graph

    def actions(self, A):
        return list(self.graph.get(A).keys())

    def result(self, state, action):
        return action

    def path_cost(self, cost, A, action, B):
        return cost + (self.graph.get(A, B) or infinity)

def depth_first_search(problem):
    global track_path
    frontier = [(Node(problem.initial))]
    explored = set()
    track_path = [problem.initial]
    while frontier:
        node = frontier.pop()
        if problem.goal_test(node.state):
            return node
        explored.add(node.state)
        expanded = node.expand(problem)
        for child in expanded:
            track_path.append(child.state)
```

```
            if child.state not in explored and child not in frontier:
                if problem.goal_test(child.state):
                    return child
                frontier.append(child)
    return None

if __name__=='__main__':
    track_path = []
    romania_problem = CityProblem(start, goal , city_map)
    node = depth_first_search(romania_problem)
    if node is not None:
        final_path = node.solution()
        final_path.insert(0, start)
        print('TRACKING PATH: ', ' -> '.join(track_path))
        print('SOLUTION PATH: ', ' -> '.join(final_path))
```

2. Jawab :
    a. Kota asal Frankfurt dan kota tujuan numberg :

```
start = 'Frankfurt'
goal = 'Numberg'

city_map = Graph(dict(
    Frankfurt = dict( Mannheim=85, Wurzburg=217, Kassel=173),
    Mannheim = dict(Karisruhe=80),
    Wurzburg= dict(Erfurt=186, Numberg=103),
    Kassel= dict(Munchen=502),
    Karisruhe = dict(Augsburg=250),
    Numberg = dict(Munchen=167, Stuttgart=183),
    Augsburg = dict(Munchen=84),
    ), directed=True)
```

    b. Cari rute solusi dan rute pelacakan :

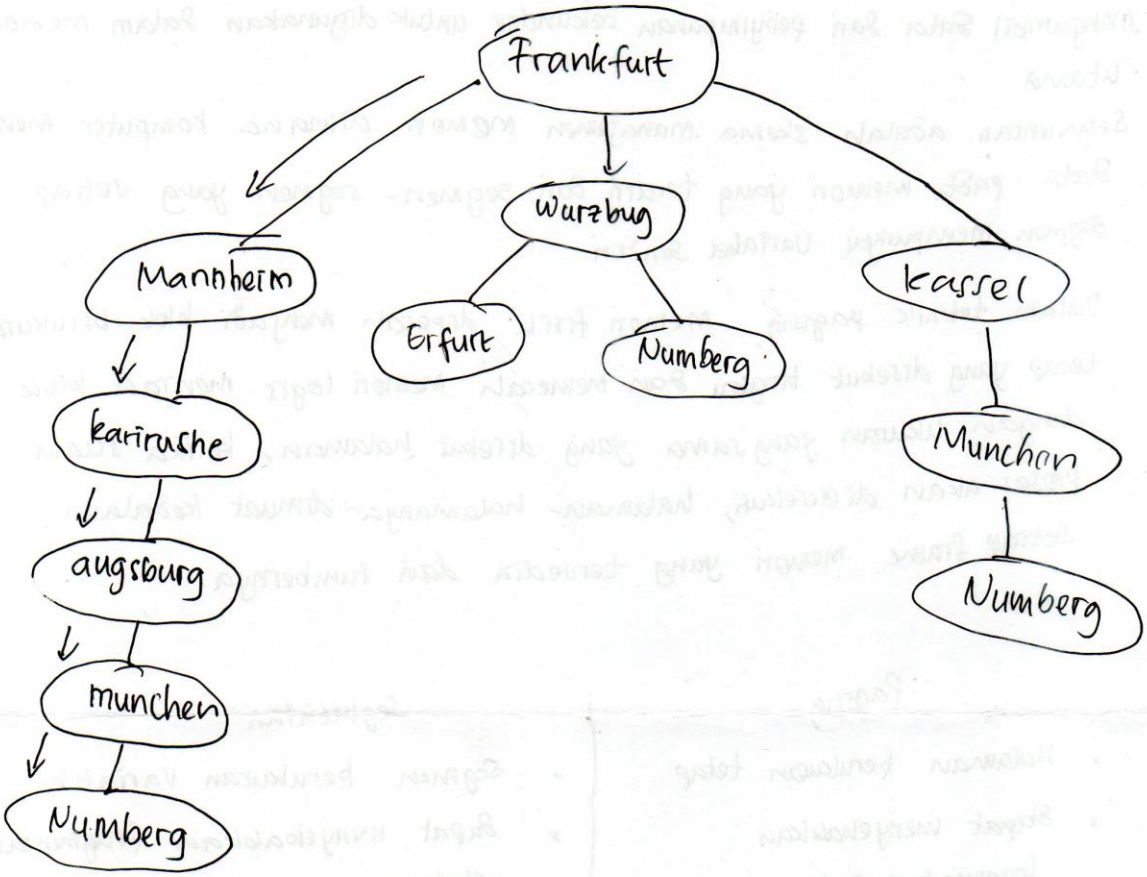
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Naufal> & "C:/Program Files/Python39/python.exe" c:/Users/Naufal/Documents/Python/Praktikum/responsi.py
TRACKING PATH:  Frankfurt -> Mannheim -> Wurzburg -> Kassel -> Munchen -> Erfurt -> Numberg
SOLUTION PATH:  Frankfurt -> Wurzburg -> Numberg
PS C:\Users\Naufal>
```

3. Gambar pohon pelacakan dan rute dengan DFS

Nama = Naufal Luthfi N

NIM = 19000182 82

Pohon Pelacalcan



Dari gambar diatas dapat dilihat bahwa rute nya adalah Frankfurt -> Mannheim -> kariruse -> Augsburg -> munchen -> numberg