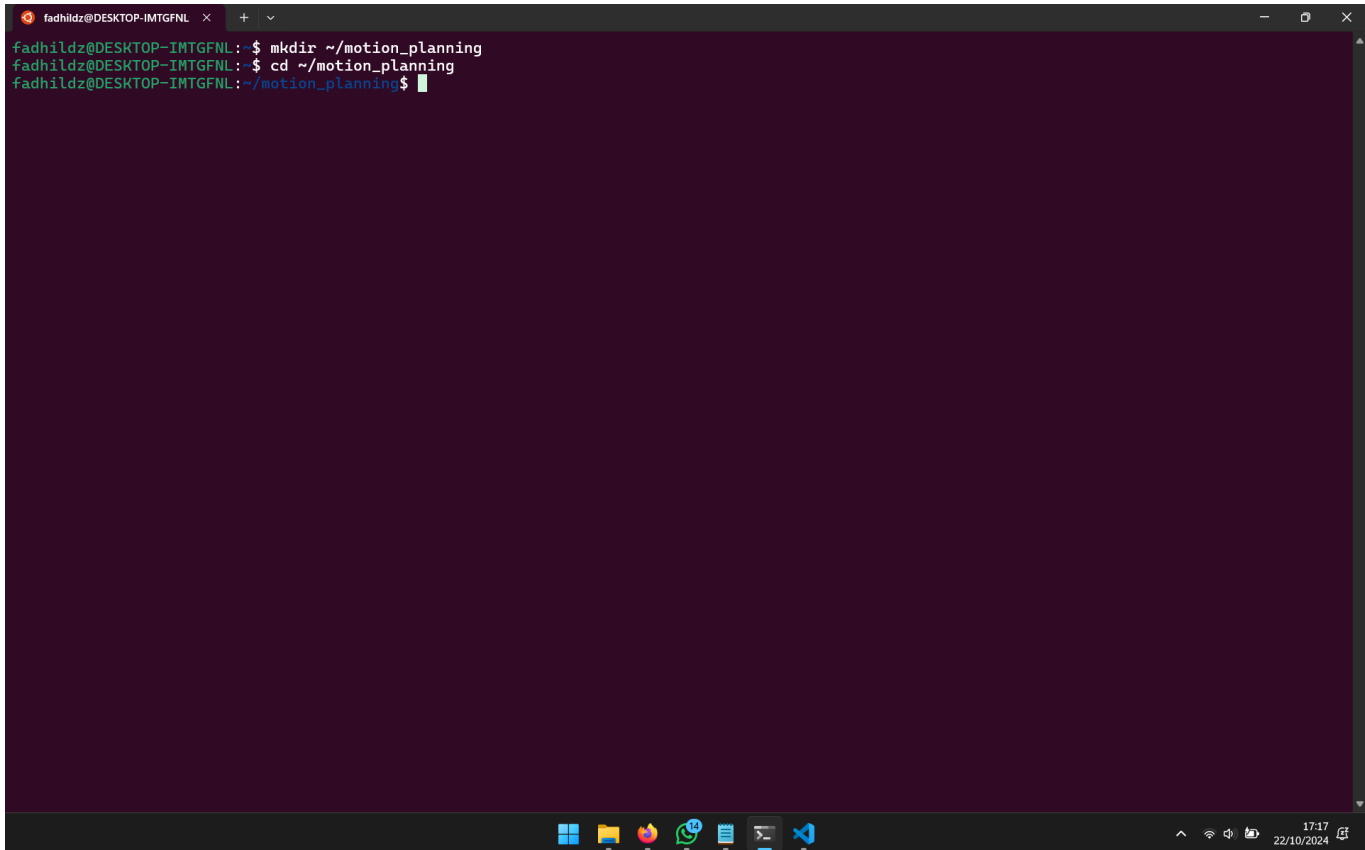


Nama : Fadhil Dzikri Aqila
NIM : 1103213136
Kelas : TK-45-G09

3 Algoritma Perencanaan Jalur

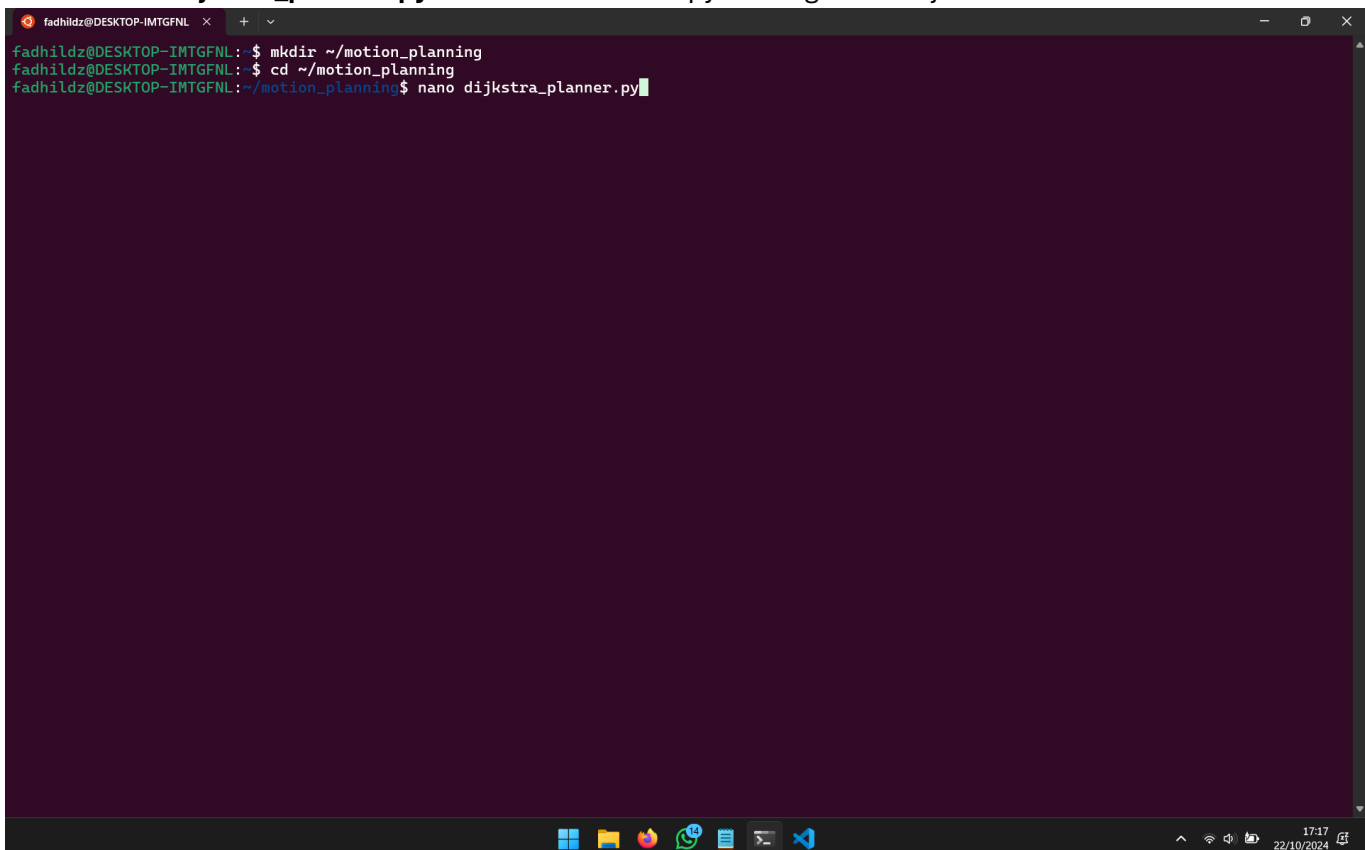
1. **mkdir ~/motion_planning** untuk membuat direktori baru dan **cd ~/motion_planning** untuk masuk ke dalam direktori yang baru dibuat.



```
fadhildz@DESKTOP-IMTGFNL: ~$ mkdir ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~$ cd ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$
```

A terminal window with a dark purple background. The prompt is 'fadhildz@DESKTOP-IMTGFNL: ~\$'. The first command 'mkdir ~/motion_planning' is entered and executed. The second command 'cd ~/motion_planning' is entered and executed. The prompt changes to 'fadhildz@DESKTOP-IMTGFNL: ~/motion_planning\$'. The taskbar at the bottom shows various application icons and the system clock displays '17:17 22/10/2024'.

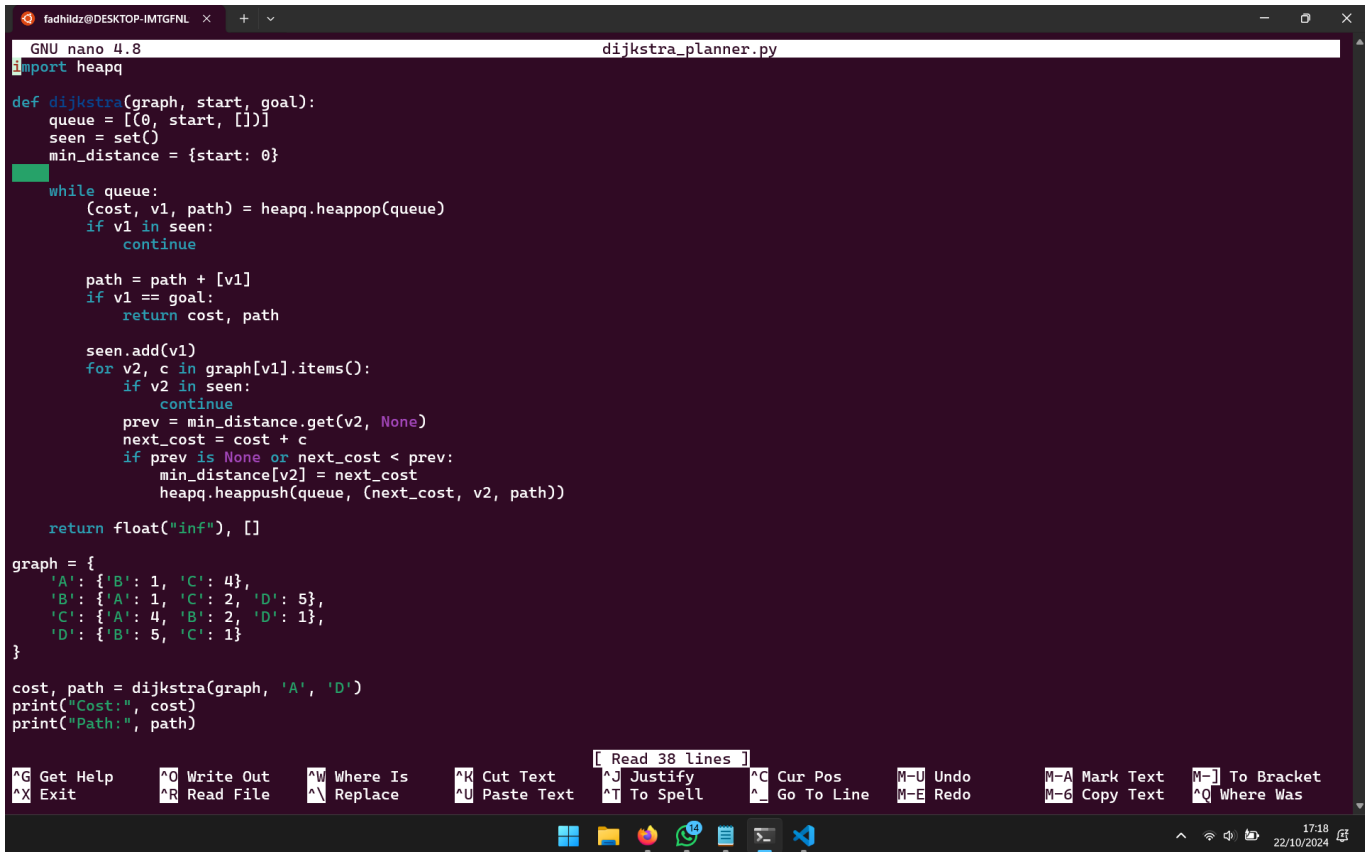
2. **nano dijkstra_planner.py** untuk membuat file python algoritma dijkstra.



```
fadhildz@DESKTOP-IMTGFNL: ~$ mkdir ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~$ cd ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ nano dijkstra_planner.py
```

A terminal window with a dark purple background. The prompt is 'fadhildz@DESKTOP-IMTGFNL: ~\$'. The first command 'mkdir ~/motion_planning' is entered and executed. The second command 'cd ~/motion_planning' is entered and executed. The prompt changes to 'fadhildz@DESKTOP-IMTGFNL: ~/motion_planning\$'. The third command 'nano dijkstra_planner.py' is entered. The taskbar at the bottom shows various application icons and the system clock displays '17:17 22/10/2024'.

3. Buat kode python untuk menjalankan algoritma dijkstra lalu save.



```
GNU nano 4.8 dijkstra_planner.py
import heapq

def dijkstra(graph, start, goal):
    queue = [(0, start, [])]
    seen = set()
    min_distance = {start: 0}

    while queue:
        (cost, v1, path) = heapq.heappop(queue)
        if v1 in seen:
            continue

        path = path + [v1]
        if v1 == goal:
            return cost, path

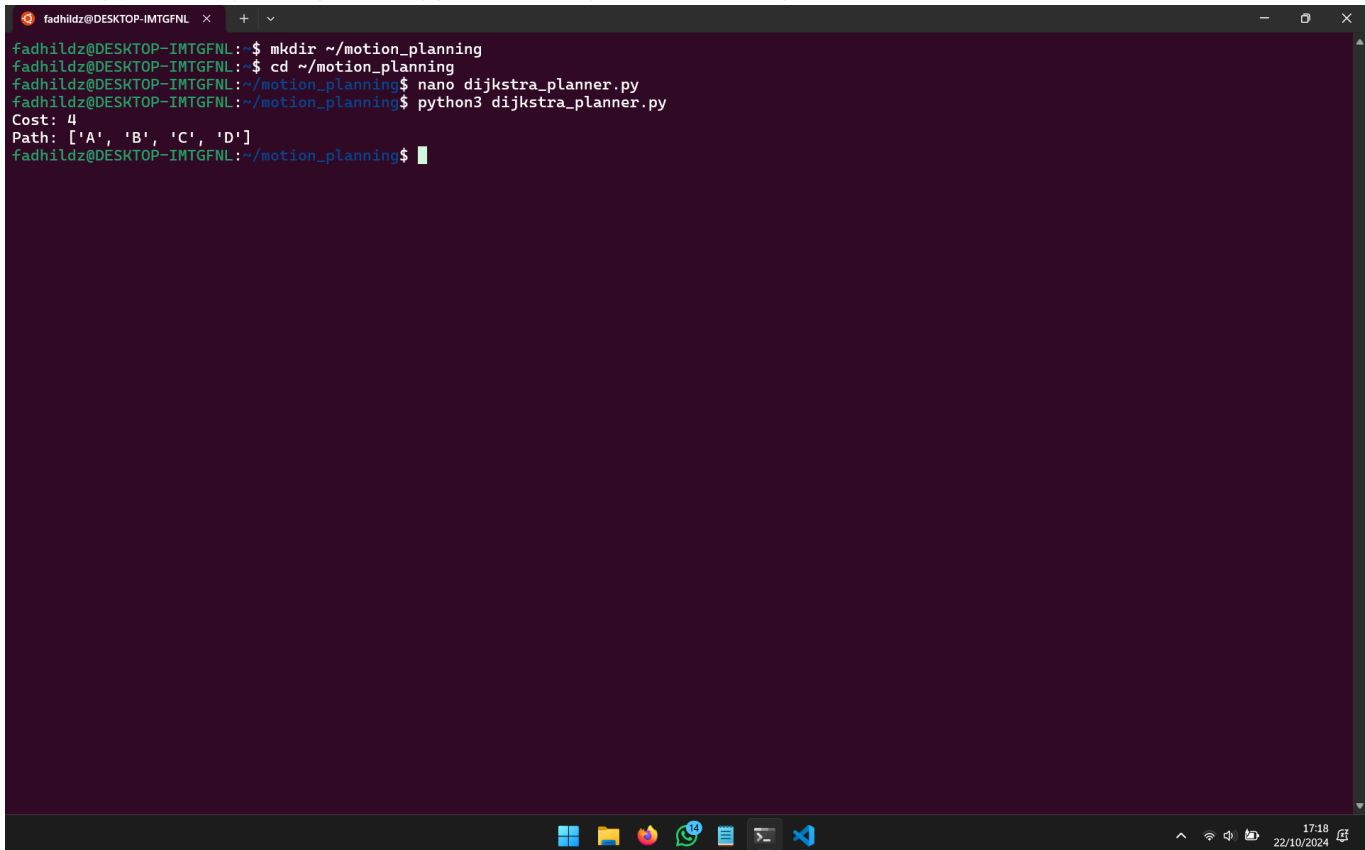
        seen.add(v1)
        for v2, c in graph[v1].items():
            if v2 in seen:
                continue
            prev = min_distance.get(v2, None)
            next_cost = cost + c
            if prev is None or next_cost < prev:
                min_distance[v2] = next_cost
                heapq.heappush(queue, (next_cost, v2, path))

    return float("inf"), []

graph = {
    'A': {'B': 1, 'C': 4},
    'B': {'A': 1, 'C': 2, 'D': 5},
    'C': {'A': 4, 'B': 2, 'D': 1},
    'D': {'B': 5, 'C': 1}
}

cost, path = dijkstra(graph, 'A', 'D')
print("Cost:", cost)
print("Path:", path)
```

4. python3 dijkstra_planner.py untuk menjalankan kodenya.



```
fadhildz@DESKTOP-IMTGFNL:~$ mkdir ~/motion_planning
fadhildz@DESKTOP-IMTGFNL:~$ cd ~/motion_planning
fadhildz@DESKTOP-IMTGFNL:~/motion_planning$ nano dijkstra_planner.py
fadhildz@DESKTOP-IMTGFNL:~/motion_planning$ python3 dijkstra_planner.py
Cost: 4
Path: ['A', 'B', 'C', 'D']
fadhildz@DESKTOP-IMTGFNL:~/motion_planning$
```

5. **nano a_star_planner.py** untuk membuat file python algoritma a*.

```
fadhildz@DESKTOP-INTGFNL: ~$ mkdir ~/motion_planning
fadhildz@DESKTOP-INTGFNL: ~$ cd ~/motion_planning
fadhildz@DESKTOP-INTGFNL: ~/motion_planning$ nano dijkstra_planner.py
fadhildz@DESKTOP-INTGFNL: ~/motion_planning$ python3 dijkstra_planner.py
Cost: 4
Path: ['A', 'B', 'C', 'D']
fadhildz@DESKTOP-INTGFNL: ~/motion_planning$ nano a_star_planner.py
fadhildz@DESKTOP-INTGFNL: ~/motion_planning$ python3 a_star_planner.py
Path: [(0, 0), (1, 0), (2, 0), (2, 1), (3, 1), (3, 2), (3, 3)]
fadhildz@DESKTOP-INTGFNL: ~/motion_planning$
```

6. Buat kode python untuk menjalankan algoritma a* lalu save.

```
GNU nano 4.8 a_star_planner.py Modified
def get_neighbors(node, grid):
    neighbors = []
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
    for d in directions:
        neighbor = (node[0] + d[0], node[1] + d[1])
        if 0 <= neighbor[0] < len(grid) and 0 <= neighbor[1] < len(grid[0]) and grid[neighbor[0]][neighbor[1]] == 0:
            neighbors.append(neighbor)
    return neighbors

def a_star(start, goal, grid):
    open_list = []
    closed_list = set()
    open_list.append(start)

    g = {start: 0}
    f = {start: heuristic(start, goal)}

    came_from = {}

    while open_list:
        current = min(open_list, key=lambda x: f[x])

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            path.append(start)
            return path[::-1]

        open_list.remove(current)
        closed_list.add(current)

        for neighbor in get_neighbors(current, grid):
            if neighbor in closed_list:
                continue

            tentative_g = g[current] + 1
            if neighbor not in open_list:
```

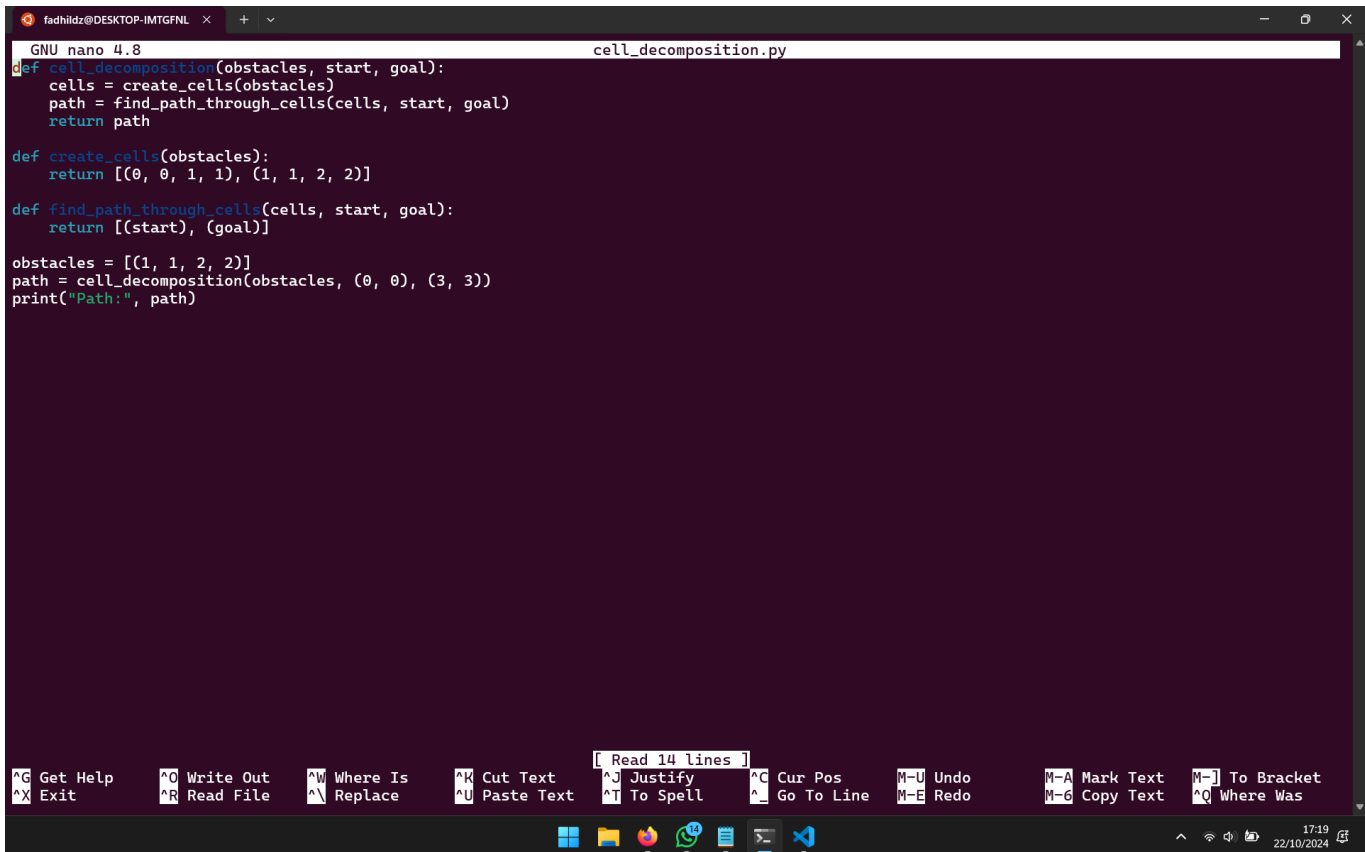
7. **python3 a_star_planner.py** untuk menjalankan kodenya.

```
fadhildz@DESKTOP-IMTGFNL: ~$ mkdir ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~$ cd ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ nano dijkstra_planner.py
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ python3 dijkstra_planner.py
Cost: 4
Path: ['A', 'B', 'C', 'D']
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ nano a_star_planner.py
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ python3 a_star_planner.py
Path: [(0, 0), (1, 0), (2, 0), (2, 1), (3, 1), (3, 2), (3, 3)]
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$
```

8. **nano cell_decomposition.py** untuk membuat file python algoritma cell decomposition.

```
fadhildz@DESKTOP-IMTGFNL: ~$ mkdir ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~$ cd ~/motion_planning
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ nano dijkstra_planner.py
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ python3 dijkstra_planner.py
Cost: 4
Path: ['A', 'B', 'C', 'D']
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ nano a_star_planner.py
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ python3 a_star_planner.py
Path: [(0, 0), (1, 0), (2, 0), (2, 1), (3, 1), (3, 2), (3, 3)]
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$ nano cell_decomposition.py
fadhildz@DESKTOP-IMTGFNL: ~/motion_planning$
```

9. Buat kode python untuk menjalankan algoritma cell decomposition lalu save.



The screenshot shows a terminal window with the GNU nano 4.8 editor open. The file being edited is `cell_decomposition.py`. The code defines three functions: `cell_decomposition`, `create_cells`, and `find_path_through_cells`. The `cell_decomposition` function calls `create_cells` and `find_path_through_cells` to find a path from a start point to a goal point, avoiding obstacles. The `create_cells` function returns a list of cells. The `find_path_through_cells` function returns a list of coordinates representing the path. The code also defines a list of obstacles and calls the `cell_decomposition` function to find a path from (0, 0) to (3, 3). The path found is [(0, 0), (3, 3)].

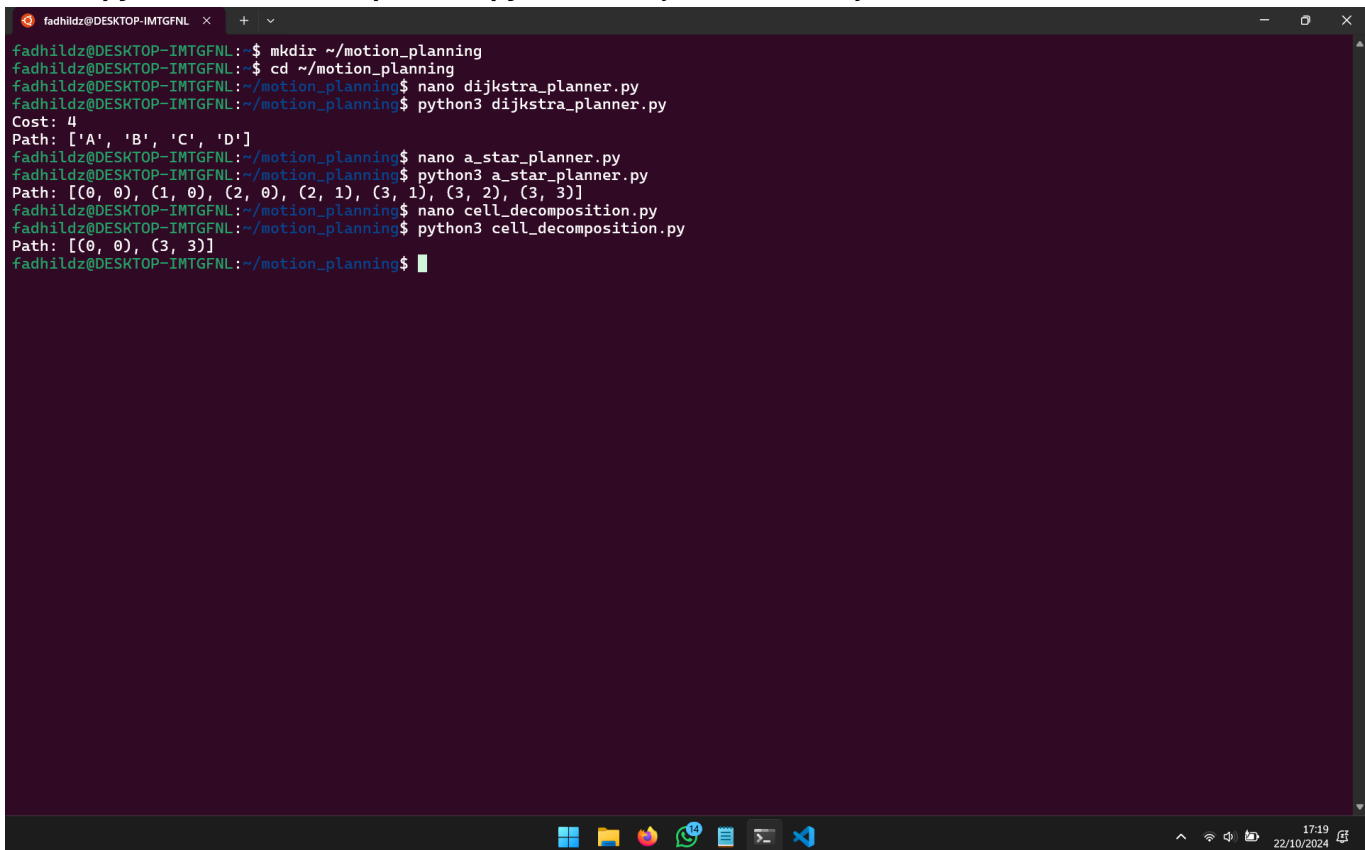
```
GNU nano 4.8 cell_decomposition.py
def cell_decomposition(obstacles, start, goal):
    cells = create_cells(obstacles)
    path = find_path_through_cells(cells, start, goal)
    return path

def create_cells(obstacles):
    return [(0, 0, 1, 1), (1, 1, 2, 2)]

def find_path_through_cells(cells, start, goal):
    return [(start), (goal)]

obstacles = [(1, 1, 2, 2)]
path = cell_decomposition(obstacles, (0, 0), (3, 3))
print("Path:", path)
```

10. python3 cell_decomposition.py untuk menjalankan kodenya.



The screenshot shows a terminal window with the following commands and output:

```
fadhildz@DESKTOP-INTGFNL:~$ mkdir ~/motion_planning
fadhildz@DESKTOP-INTGFNL:~$ cd ~/motion_planning
fadhildz@DESKTOP-INTGFNL:~/motion_planning$ nano dijkstra_planner.py
fadhildz@DESKTOP-INTGFNL:~/motion_planning$ python3 dijkstra_planner.py
Cost: 4
Path: ['A', 'B', 'C', 'D']
fadhildz@DESKTOP-INTGFNL:~/motion_planning$ nano a_star_planner.py
fadhildz@DESKTOP-INTGFNL:~/motion_planning$ python3 a_star_planner.py
Path: [(0, 0), (1, 0), (2, 0), (2, 1), (3, 1), (3, 2), (3, 3)]
fadhildz@DESKTOP-INTGFNL:~/motion_planning$ nano cell_decomposition.py
fadhildz@DESKTOP-INTGFNL:~/motion_planning$ python3 cell_decomposition.py
Path: [(0, 0), (3, 3)]
fadhildz@DESKTOP-INTGFNL:~/motion_planning$
```