

# Tugas Besar

# Strategi Algoritma

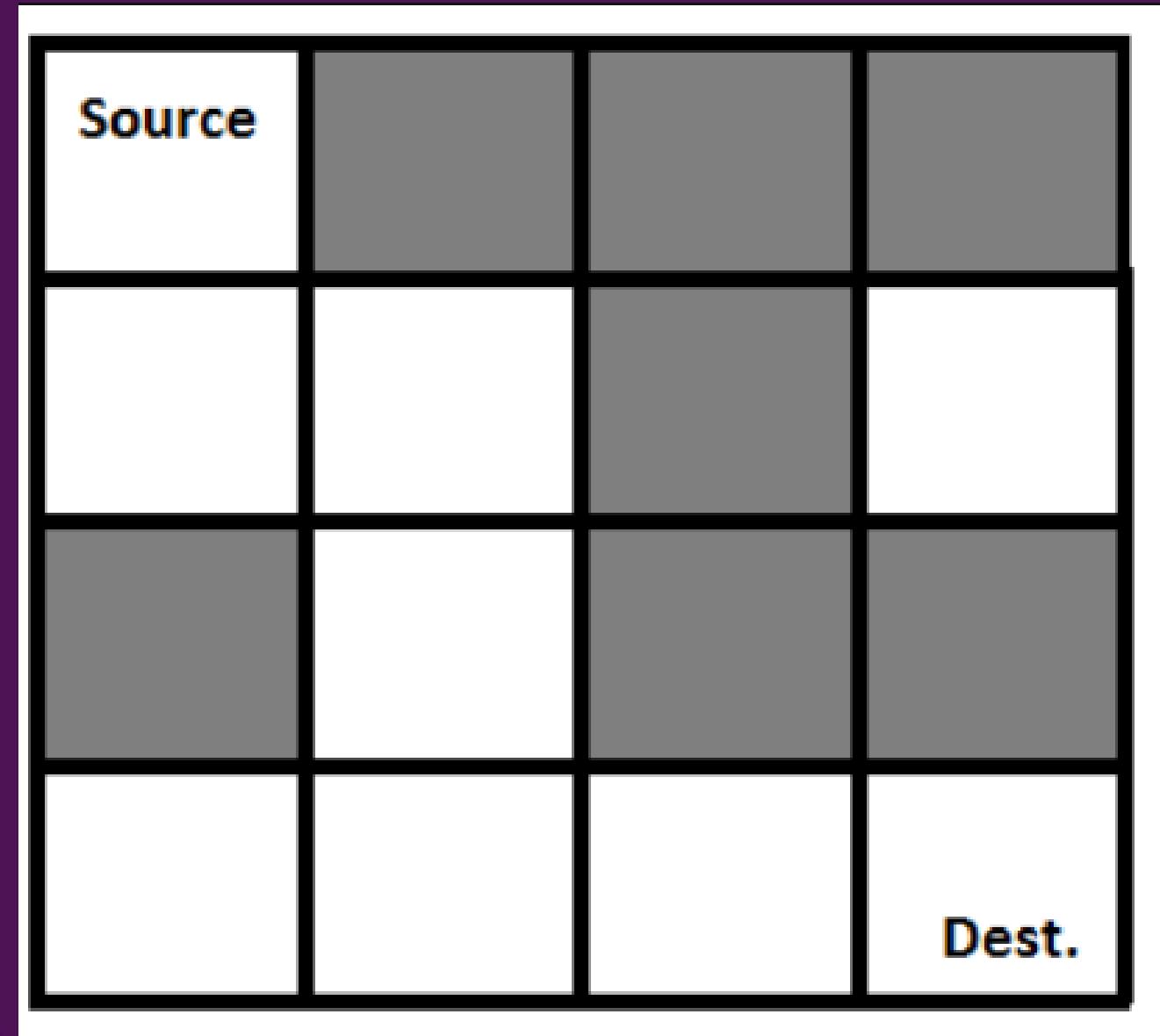
Kelompok 1 IF-45-05

1. Hanrocky Halim 1301213446
2. Prins Naval Nuzen 1301213097
3. Naufal Hilmi Majid 1301213430

# TOPIK :

## Rat In A Maze:

Terdapat sebuah maze berukuran  $N \times M$  tikus akan mulai dari starting point menuju end point.



# ALGORITMA:

1. **Best Fist Search**: algoritma pencarian yang memilih langkah berikutnya berdasarkan penilaian heuristik yang menjanjikan secara relatif.
2. **Backtracking** : pendekatan rekursif yang secara sistematis mencoba setiap kemungkinan solusi

# 1. Best Fist Search

```
def _calculate_heuristic(self, current, end):
    current_row, current_col = current
    end_row, end_col = end
    return abs(end_row - current_row) + abs(end_col - current_col)
```

```
def _best_first_search(self, start, end):
    queue = PriorityQueue()
    queue.put((0, start))
    visited = set([start])

    while not queue.empty():
        cost, current = queue.get()
        current_row, current_col = current

        if current == end:
            self._build_solution()
            return True

        neighbors = self._get_neighbors(current_row, current_col)

        for neighbor in neighbors:
            neighbor_row, neighbor_col = neighbor
            if neighbor not in visited:
                priority = self._calculate_heuristic(neighbor, end)
                queue.put((priority, neighbor))
                visited.add(neighbor)
                self.solution[neighbor_row][neighbor_col] = '*'
                self.step_count += 1

    return False
```

Pilih sumber masukan:

1. Masukkan labirin melalui keyboard
2. Baca labirin dari file teks

Pilihan: 2

Masukkan path file: maze.txt

```
# # # # # # # # # # # #
# S * * # * * * # # # # #
# * # * # * * * #   #   # #
#   # * # * # * # *   # # #
#   # * * * # * * * #   # #
#   # # # # # * # * # * # #
#   #           # * * * # #
#   # # # # #   # # # # * # #
#           * * E #
# # # # # # # # # # # #
Langkah yang dibutuhkan: 31
Waktu yang dibutuhkan: 0.003906799998731003 detik
```

## 2. Backtracking

```
def _backtrack(self, current, end, solution):
    if current == end:
        self.solution = solution
        return True

    current_row, current_col = current

    if self._is_valid_move(current_row, current_col):
        solution.append(current)
        self.step_count += 1

        # Check right
        next_position = (current_row, current_col + 1)
        if next_position not in solution:
            if self._backtrack(next_position, end, solution):
                return True

        # Check down
        next_position = (current_row + 1, current_col)
        if next_position not in solution:
            if self._backtrack(next_position, end, solution):
                return True

        # Check up
        next_position = (current_row - 1, current_col)
        if next_position not in solution:
            if self._backtrack(next_position, end, solution):
                return True

        # Check left
        next_position = (current_row, current_col - 1)
        if next_position not in solution:
            if self._backtrack(next_position, end, solution):
                return True

    return False
```

1. Masukkan labirin melalui keyboard
2. Baca labirin dari file teks

Pilihan: 2

Masukkan nama file: maze.txt

Solusi Backtracking:

```
# # # # # # # # # # # #
# S 0 0 #      # # # # # #
#   # 0 # 0 0 0 #   #   # #
#   # 0 # 0 # 0 #   #   # #
#   # 0 0 0 # 0 0 0 #   # #
#   # # # # #   # 0 # #   # #
#   #           # 0 0 0 0 # #
#   # # # # #   # # # # 0 # #
#                   0 E #
# # # # # # # # # # # # #
```

Langkah yang dibutuhkan: 23

Waktu yang dibutuhkan: 0.011603000000832253 detik

# KOMPLEKSITAS:

1. **Best Fist Search:**  $O(N + M \log N)$
2. **Backtracking :**  $O(4^N)$

# **SKEMA PENELITIAN:**

**penelitian dilakukan dengan menggunakan empat skema yang berbeda.** Skema-skema tersebut didasarkan pada kombinasi dua parameter, **yaitu parameter ukuran rintangan (size obstacle) dan tipe rintangan (tipe obstacle).**

# SKEMA 1:

10x14

```
#####
#S   #   #####
#  #  #   #  #  #####
#  #  #  #  #   #####
#  #    #  #  ##
#  ##### #  ##  ##
#  #      #      ##
#  ##### #  #####
#  #      #      ##
#      E#
#####
```

## 1. Best Fist Search:

running time: 0,00351

Langkah: 31

RT/L: 0,00011329

## 2. Backtracking : O(4<sup>n</sup>)

running time: 0,00407

Langkah: 23

RT/L: 0,000176913

# SKEMA 2:

10x14

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### |
| #S    | #     |       |       |       |       |       |       |       |       |       |       |       |       |       |
| #     | #     | #     |       | ###   | ###   | ###   |       |       |       |       |       |       |       |       |
| #     | #     | #     | #     | #     | #     |       | ###   |       |       |       |       |       |       |       |
| #     | #     | #     | #     | #     | #     |       |       |       |       |       |       |       |       |       |
| #     | #     |       | #     | #     | #     |       |       |       |       |       |       |       |       |       |
| #     | ####  | #     | ####  | ##    |       |       |       |       |       |       |       |       |       |       |
| #     |       |       | #     |       |       |       |       |       |       |       |       |       |       |       |
| ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### |
| #     |       |       |       |       |       |       | E#    |       |       |       |       |       |       |       |
| ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### | ##### |

## 1. Best Fist Search:

running time: 0,01042

Langkah: 37

RT/L: 0,000281703

## 2. Backtracking : $O(4^n)$

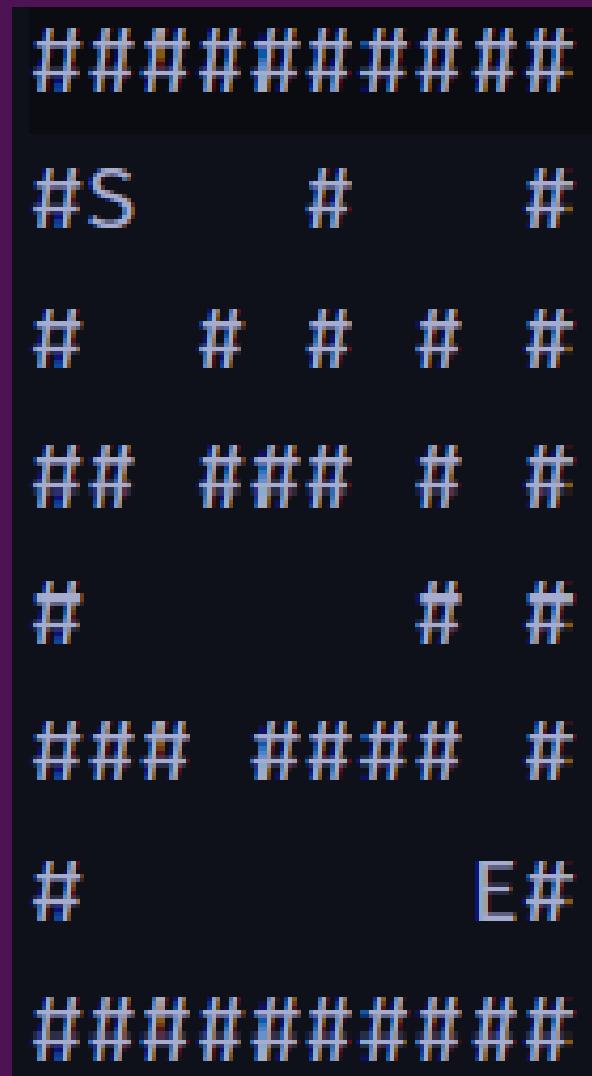
running time: 0,03061

Langkah: 32

RT/L: 0,000956563

# SKEMA 3:

10x8



## 1. Best Fist Search:

running time: 0,00455

Langkah: 24

RT/L: 0,000189667

## 2. Backtracking : O(4^n)

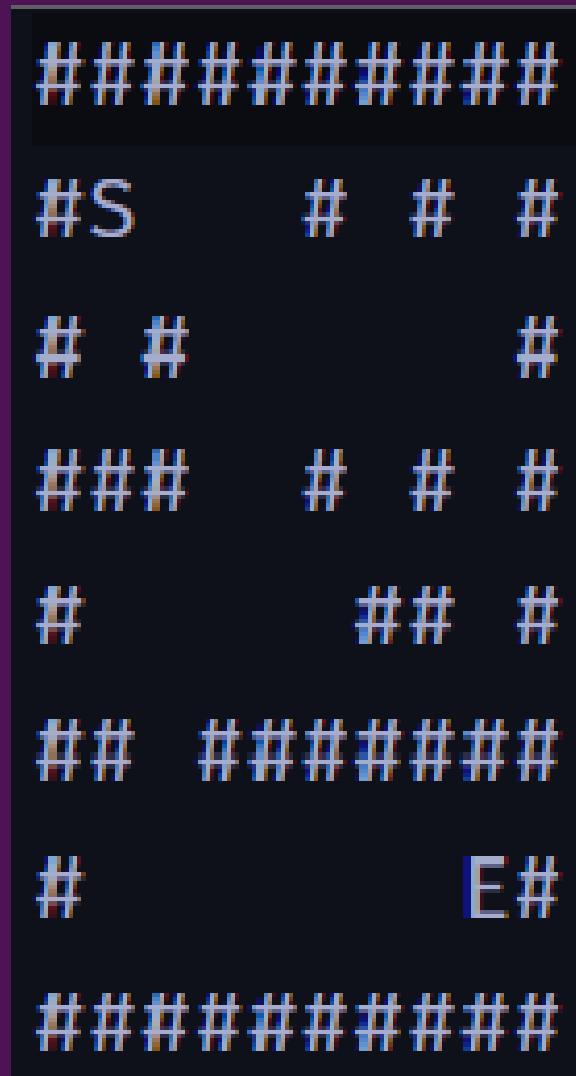
running time: 0,00538

Langkah: 21

RT/L: 0,000255952

# SKEMA 4 :

10x8



## 1. Best Fist Search:

running time: 0,00358

Langkah: 31

RT/L: 0,000115355

## 2. Backtracking : O(4<sup>8</sup>)

running time: 0,00345

Langkah: 28

RT/L: 0,000123107



# Kesimpulan

Algoritma Best First Search (BFS) secara konsisten memiliki running time yang lebih rendah dibandingkan dengan algoritma Backtracking.

**TERIMA KASIH**