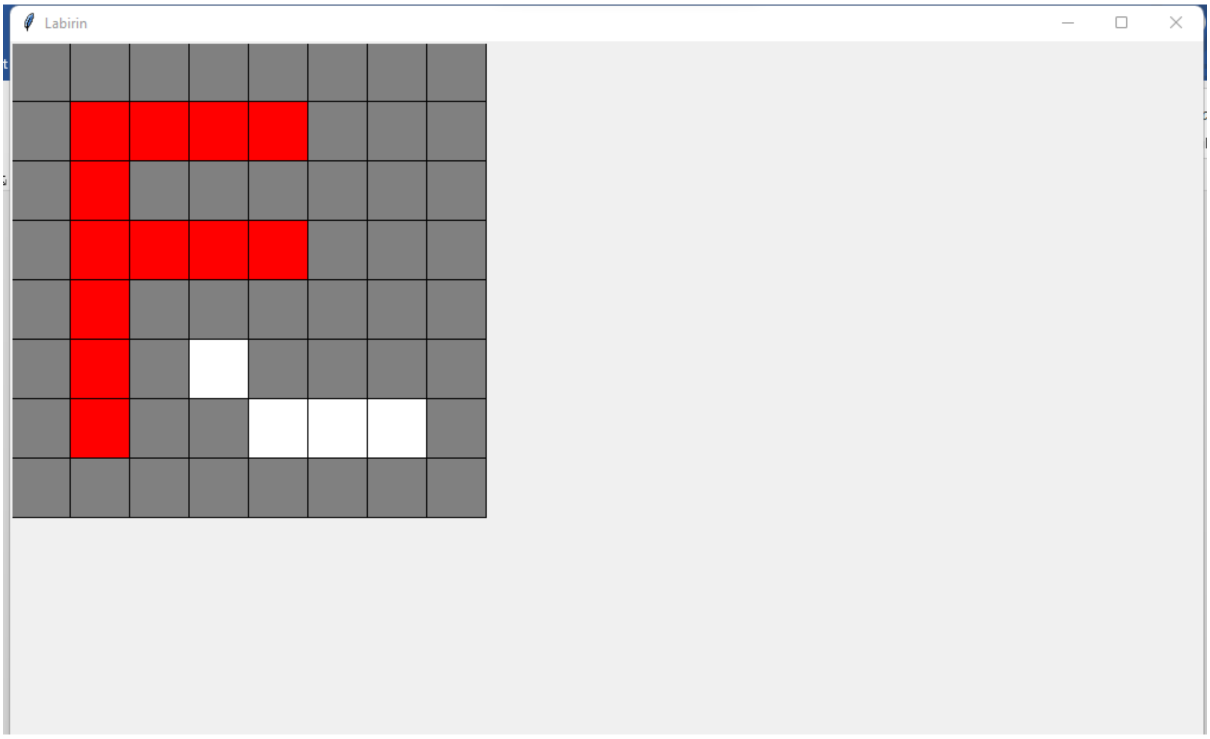
Nama: Rohmatulloh Fadhilah

NIM: 212410103026

Kelas : Algo 2 E

Hasil program



Code Program

```
import tkinter as tk


window = tk.Tk()

window.title('Labirin')

cnv = tk.Canvas(window, height=1000, width=1000)

cnv.pack()


inputan = input()

panjang_baris = int(inputan.split(' ')[0])

lebar_baris = int(inputan.split(' ')[1])
```

```
hasil_base = []
# mengumpulkan array ke dalam hasil_base
for i in range(lebar_baris):
    baris = input()
    base_temp = []
    for j in range(panjang_baris):
        if baris[j] == '0':
            base_temp.append(0)
        else:
            base_temp.append(1)
    hasil_base.append(base_temp)

# menyusun hasil_base menjadi struktur labirin
def create_base(base):
    for i in base:
        for j in i:
            print(j, end='')
        print()

def baseSolver(base, linePosition, finishPosition):
    line_y, line_x = linePosition
    finish_x, finish_y = finishPosition
    # kotak pertama diberi nomor 2
    base[line_y][line_x] = 2

    # cek jalur atas
    if base[line_y-1][line_x] == 0:
        # jika jalur atas sudah buntu dan harus kembali
        mustReturn = baseSolver(base, [line_y-1, line_x], finishPosition)
        if mustReturn:
            return True
    elif base[line_y-1][line_x] == base[finish_y][finish_x]:
        base[line_y][line_x] = 4
        return True
```

```
# cek jalur bawah
if base[line_y+1][line_x] == 0:
    # jika jalur bawah sudah buntu dan harus kembali
    mustReturn = baseSolver(base, [line_y+1, line_x], finishPosition)
    if mustReturn:
        return True
elif base[line_y+1][line_x] == base[finish_y][finish_x]:
    base[line_y][line_x] = 4
    return True

# cek jalur kiri
if base[line_y][line_x-1] == 0:
    # jika jalur kiri sudah buntu dan harus kembali
    mustReturn = baseSolver(base, [line_y, line_x-1], finishPosition)
    if mustReturn:
        return True
elif base[line_y][line_x-1] == base[finish_y][finish_x]:
    base[line_y][line_x] = 4
    return True



# cek jalur kanan
if base[line_y][line_x+1] == 0:
    mustReturn = baseSolver(base, [line_y, line_x+1], finishPosition)
    if mustReturn:
        return True
elif base[line_y][line_x+1] == base[finish_y][finish_x]:
    base[line_y][line_x] = 4
    return True


base[line_y][line_x] = 3
```

```
# letak robot
robot_position = input()
robot_position = int(robot_position.split(' ')[1]), int(robot_position.split(' ')[0])

# letak objek
object_position = input()
object_position = int(object_position.split(' ')[0]), int(object_position.split(' ')[1])

print(baseSolver(hasil_base, robot_position, object_position))
create_base(hasil_base)

# menggambar di tkinter berdasarkan hasil_base yang baru
for i in range(lebar_baris):
    for j in range(panjang_baris):
        if hasil_base[i][j] == 0:
            cnv.create_text(j*50+25, i*50+25, text='0', fill='white')
            cnv.create_rectangle(j*50, i*50, (j+1)*50, (i+1)*50, fill='white')
        elif hasil_base[i][j] == 1:
            cnv.create_text(j*50+25, i*50+25, text='0', fill='white')
            cnv.create_rectangle(j*50, i*50, (j+1)*50, (i+1)*50, fill='grey')
        elif hasil_base[i][j] == 2:
            cnv.create_text(j*50+25, i*50+25, text='0', fill='white')
            cnv.create_rectangle(j*50, i*50, (j+1)*50, (i+1)*50, fill='yellow')
        elif hasil_base[i][j] == 3:
            cnv.create_text(j*50+25, i*50+25, text='0', fill='white')
            cnv.create_rectangle(j*50, i*50, (j+1)*50, (i+1)*50, fill='red')
        elif hasil_base[i][j] == 4:
            cnv.create_text(j*50+25, i*50+25, text='0', fill='white')
            cnv.create_rectangle(j*50, i*50, (j+1)*50, (i+1)*50, fill='green')

window.mainloop()
```