

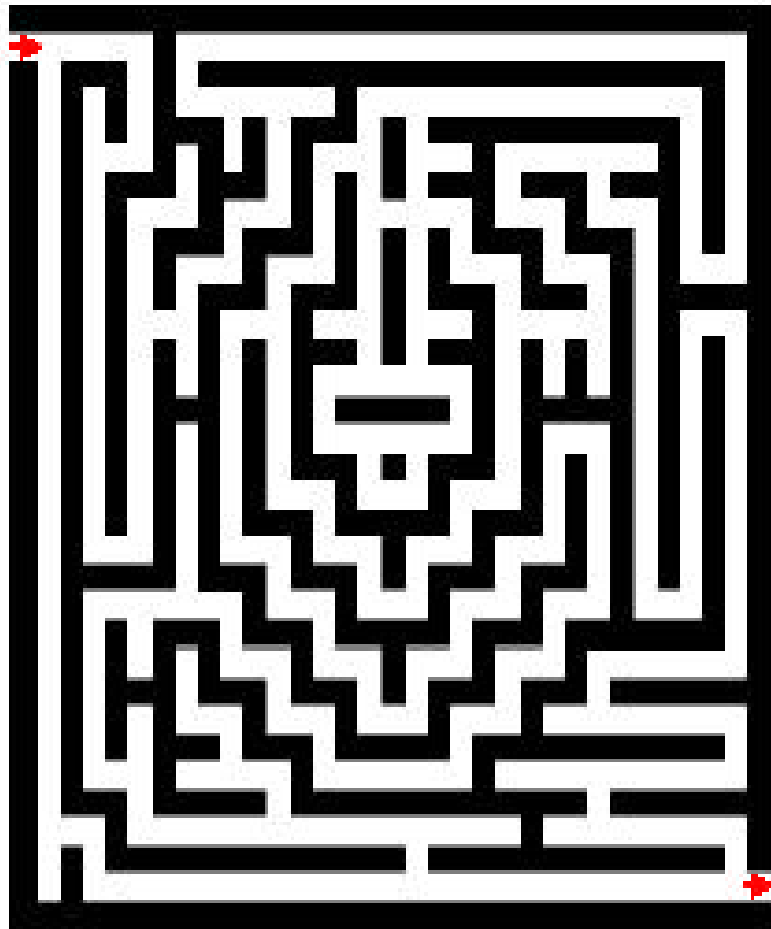
Algoritma Runut-balik ***(Backtracking)* – Bag. 2**

Bahan Kuliah IF2251 Strategi Algoritmik

Oleh: Rinaldi Munir

Aplikasi Algoritma Runut-balik untuk *Games*

Mencari jalan keluar di dalam labirin (*Maze Problem*).



Penyelesaian dengan *backtracking*:

- Bagi lintasan menjadi sederetan langkah. Sebuah langkah terdiri dari pergerakan satu unit sel pada arah tertentu.
- Arah yang mungkin: ke atas (*up*), ke bawah (*down*), ke kiri (*left*), ke kanan (*right*).

Garis besar algoritma runut-baliknya:

```
while belum sampai pada tujuan do  
  if terdapat arah yang benar sedemikian sehingga kita belum pernah  
    berpindah ke sel pada arah tersebut  
  then  
    pindah satu langkah ke arah tersebut  
  else  
    backtrack langkah sampai terdapat arah seperti yang disebutkan  
    di atas  
  endif  
endwhile
```

- Bagaimana mengetahui langkah yang mana yang perlu dijejaki kembali?
- Ada dua solusi untuk masalah ini: pertama, simpan semua langkah yang pernah dilakukan, atau kedua, gunakan rekursi (yang secara implisit menyimpan semua langkah).
- Rekursi adalah solusi yang lebih mudah.

```
function SolveMaze(input M: labirin) →boolean  
{ true jika solusi ditemukan, false jika tidak }
```

Deklarasi

```
arah : integer { up = 1, down, 2, left = 3, right = 4 }
```

Algoritma:

```
if solusi sudah ditemukan then  
    return true
```

```
else
```

```
    for tiap arah gerakan (up, down, left, right) do
```

```
        move(M arah) { pindah satu langkah (satu sel) sesuai arah tersebut }
```

```
        if SolveMaze(M) then
```

```
            return true
```

```
        else
```

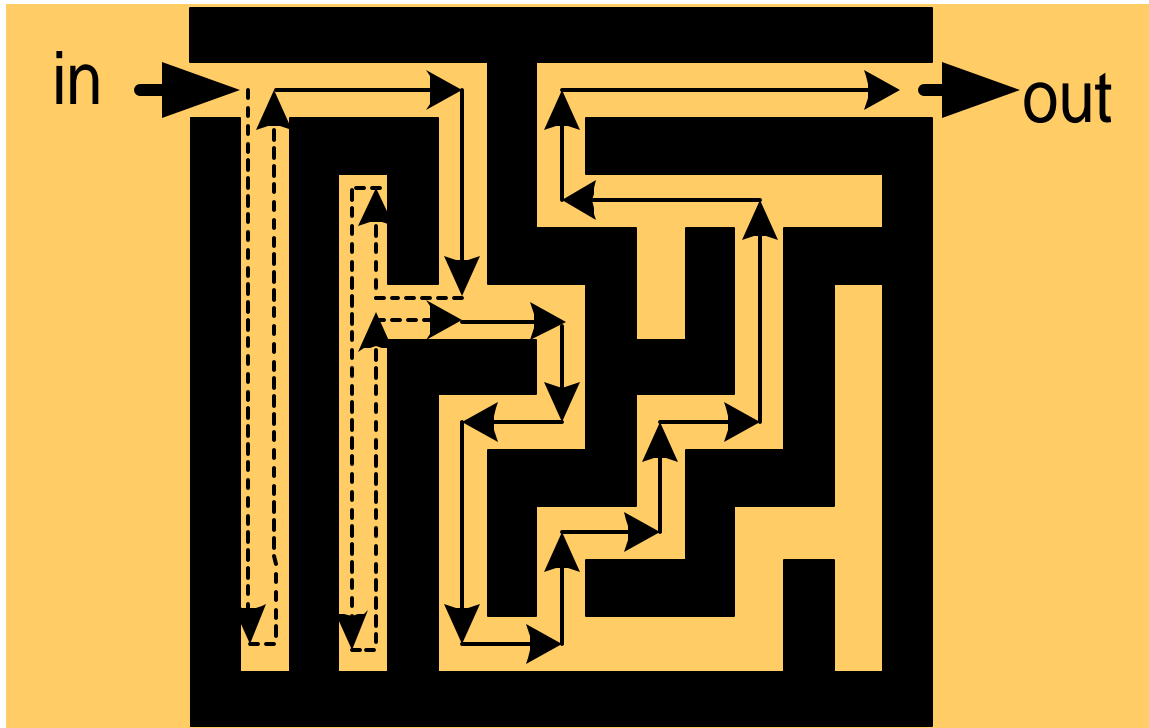
```
            unmove(M arah) { backtrack }
```

```
        endif
```

```
    endfor
```

```
    return false { semua arah sudah dicoba, tetapi tetap buntu,  
                    maka  
                    kesimpulannya: tidak ada solusi }
```

```
endif
```



Contoh runut-balik pada sebuah labirin. Runut-balik diperlihatkan dengan garis putus-putus.

Contoh lain:

