

# Tugas Individu Kuliah Asinkron RPL

Saturday, 05 December 2020 19.53

R. Komang Eling Pramuatmaja (185150200111081) | Kode di <https://github.com/elingp/tugas-rpl-testing>

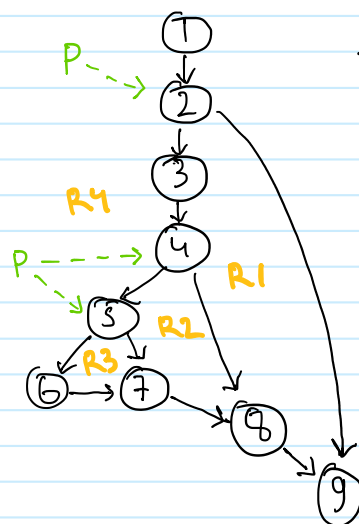
Selection Sort

```
procedure PengurutanSeleksi(input/output s : array [1..n] of integer)
{ Mengurutkan s1, s2, ..., sn sehingga tersusun menaik dengan metode pengurutan seleksi.
Masukan: s1, s2, ..., sn
Keluaran: s1, s2, ..., sn (terurut menaik)
}
Deklarasi
i, j, inaks, temp : integer
Algoritma:
for i ← n downto 2 do { jumlah pass sebanyak n - 1 }
{ cari elemen terbesar di dalam s[1], s[2], ..., s[i] }
inaks ← 1 { elemen pertama diasumsikan sebagai elemen terbesar sementara }
for j ← 2 to i do
if s[j] > s[inaks] then
inaks ← j
endif
endfor
{pertukarkan s[inaks] dengan s[i] }
temp ← s[i]
s[i] ← s[inaks]
s[inaks] ← temp
endfor
```

Handwritten annotations on the code:

- ① points to the start of the algorithm.
- ② points to the first pass of the outer loop.
- ③ points to the inner loop.
- ④ points to the comparison condition.
- ⑤ points to the update of inaks.
- ⑥ points to the swap operation.
- ⑦ points to the end of the inner loop.
- ⑧ points to the swap operation.
- ⑨ points to the end of the outer loop.

## Flow Graph



### Nilai V(G)

•  $V(G) = \text{jumlah region}$   
 $= 4$  ✓

•  $V(G) = E - N + 2$   
 $= 11 - 9 + 2$   
 $= 4$  ✓

•  $V(G) = P + 1$   
 $= 3 + 1$   
 $= 4$  ✓

### Jalur independen

• Jalur 1: 1-2-9

• Jalur 2: 1-2-3-4-8-9

• Jalur 3: 1-2-3-4-5-7-8-9

• Jalur 4: 1-2-3-4-5-6-7-8-9

### Kasus uji :

- Jalur 1 : array of integer dengan panjang 1  
(e.g.  $s = \{1\}$ )
- Jalur 2 : inisialisasi j dengan nilai lebih besar daripada i  
(e.g.  $j = i + 1$ )
- Jalur 3 : array of integer sedemikian sehingga indeks yang disimpan oleh inaks pertama kali selalu integer dengan nilai terbesar dari sisa iterasi i  
(e.g.  $s = \{10, 8, 9\}$ )

- Jalur 4 : array of integer yang urut secara menaik  
(e.g.  $s = \{1, 2\}$ )

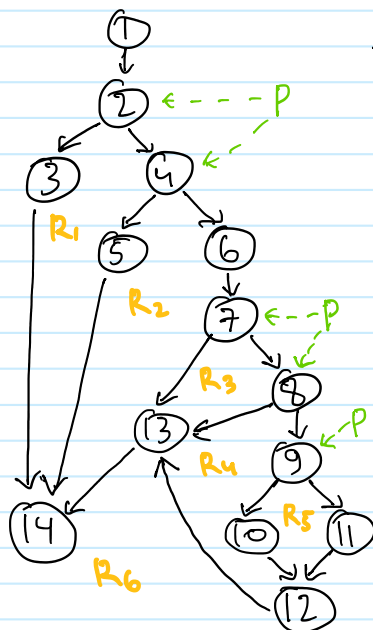
```
function Prima(input x : integer) → boolean
{ Menguji apakah x bilangan prima atau bukan.
  Masukan: x
  Keluaran: true jika x prima, atau false jika x tidak prima.
}
```

**Deklarasi**

```
k, y : integer } ①
test : boolean }
```

**Algoritma:**

```
if x < 2 then { 1 bukan prima }
  return false → ③
else
  if x = 2 then { 2 adalah prima, kasus khusus }
    return true → ⑤
  else
    y ← ⌊√x⌋ → ⑥
    test ← true → ⑦
    while (test) and (y ≥ 2) do
      if x mod y = 0 then → ⑧
        test ← false → ⑩
      else
        y ← y - 1 → ⑪
      endif
    endwhile → ⑫
    { not test or y < 2 }
  return test → ⑬
endif → ⑭
endif
```



Nilai  $V(G)$

•  $V(G) = \text{jumlah region}$   
= 6 ✓

•  $V(G) = E - N + 2$   
= 18 - 14 + 2  
= 6 ✓

•  $V(G) = P + 1$   
= 5 + 1  
= 6 ✓

Jalur independen

•  $J_1 : 1 - 2 - 3 - 14$

•  $J_2 : 1 - 2 - 4 - 5 - 14$

•  $J_3 : 1 - 2 - 4 - 6 - 7 - 13 - 14$

•  $J_4 : 1 - 2 - 4 - 6 - 8 - 13 - 14$

•  $J_5 : 1 - 2 - 4 - 6 - 8 - 9 - 10 - 12 - 13 - 14$

•  $J_6 : 1 - 2 - 4 - 6 - 8 - 9 - 11 - 12 - 13 - 14$

Kasus uji

- $J_1$  : nilai masukan  $\times$  kurang dari 2  
(e.g.  $x = 1$ )

- $J_2$ : nilai masukan  $x$  sama dengan 2  
(e.g.  $x = 2$ )
- $J_3$ : inisialisasi test dengan nilai false  
(e.g. test = false)
- $J_4$ : inisialisasi  $y$  dengan nilai kurang dari 2  
(e.g.  $y = 1$ )
- $J_5$ : nilai masukan  $x$  adalah bilangan bukan prima lebih dari 2  
(e.g.  $x = 4$ )
- $J_6$ : nilai masukan  $x$  adalah bilangan prima lebih dari 2  
(e.g.  $x = 3$ )

```

procedure PencocokanString(input P : string, T : string,
                           n, m : integer, output idx : integer)
{ Masukan: pattern P yang panjangnya m dan teks T yang
  panjangnya n. Teks T direpresentasikan sebagai string
  (array of character)
  Keluaran: lokasi awal kecocokan (idx)
}
Deklarasi
  i : integer
  ketemu : boolean

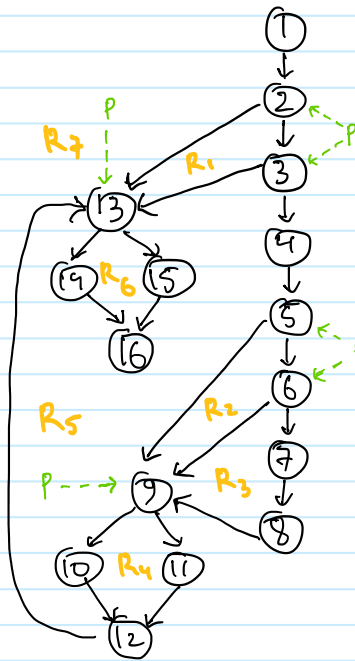
```

#### Algoritma:

```

① i ← 0
   { ketemu ← false
     while (i ≤ n-m) and (not ketemu) do
       j ← 1
       while (j ≤ m) and (Pj = Ti+j) do
         j ← j+1
       endwhile
       { j > m or Pj ≠ Ti+j }
       if j = m then { kecocokan string ditemukan }
         ketemu ← true
       else
         i ← i+1 { geser pattern satu karakter ke kanan teks }
       endif
     endwhile
   }
   { i > n - m or ketemu }
   if ketemu then
     idx ← i+1
   else
     idx ← -1
   endif

```



### Nilai $V(G)$

$$\bullet V(G) = \text{jumlah region} = 7$$

$$\bullet V(G) = E - N + 2 = 21 - 16 + 2 = 7$$

$$\bullet V(G) = P + 1 = 6 + 1 = 7$$

### Jalur independen

$$\bullet J_1: 1-2-13-15-16$$

$$\bullet J_2: 1-2-13-14-16$$

$$\bullet J_3: 1-2-3-13-14-16$$

$$\bullet J_4: 1-2-3-4-5-9-11-12-13-15-16$$

$$\bullet J_5: 1-2-3-4-5-6-9-11-12-13-15-16$$

$$\bullet J_6: 1-2-3-4-5-6-7-8-9-11-12-13-15-16$$

$$\bullet J_7: 1-2-3-4-5-6-7-8-9-10-12-13-14-16$$

### Kasus uji

•  $J_1$ : inisialisasi  $i$  dengan nilai lebih dari  $n-m$   
(e.g.  $n=10, m=4, i=7$ )

•  $J_2$ : inisialisasi  $i$  dengan nilai lebih dari  $n-m$  dan ketemu dengan nilai true  
(e.g.  $n=10, m=4, i=7$ , ketemu = true)

•  $J_3$ : inisialisasi ketemu dengan nilai true  
(e.g. ketemu = true)

•  $J_4$ : inisialisasi  $j$  dengan nilai lebih besar dari  $m$  dan  $i$  sama dengan  $n-m$   
(e.g.  $n=10, m=4, i=6, j=5$ )

•  $J_5$ : masukan  $P$  dan  $T$  sama sekali tidak memiliki kesamaan  
(e.g.  $P = "bbbb"$ ,  $T = "hahahahaha"$ )

•  $J_6$ : masukan  $P$  memiliki sedikit kesamaan dengan teks  $T$   
(e.g.  $P = "he"$ ,  $T = "hahahahaha"$ )

•  $J_7$ : masukan  $P$  memiliki kesamaan dengan teks  $T$   
(e.g.  $P = "haha"$ ,  $T = "hahahahaha"$ )