

Tugas Conditional Expressions

Elisia Dwi Rahayu

2022-09-24

```
library(dslabs)
data("murders")
```

Soal Nomor 1

Fungsi `nchar` dapat digunakan untuk menghitung jumlah karakter dari suatu vektor karakter. Buatlah satu baris kode yang akan menyimpan hasil komputasi pada variabel `'new_names'` dan berisi singkatan nama negara ketika jumlah karakternya lebih dari 8 karakter.

```
count <- nchar(murders$state)
new_names <- count
total <- length(murders$state)

for(i in 1:total){
  if(count[i] < 8)
  {
    new_names[i] <- count[i]
  } else
  {
    new_names[i] <- murders$abb[i]
  }
}

new_names

## [1] "7" "6" "7" "AR" "CA" "CO" "CT" "DE" "DC" "7" "7" "6" "5" "IL" "7"
## [16] "4" "6" "KY" "LA" "5" "MD" "MA" "MI" "MN" "MS" "MO" "7" "NE" "6" "NH"
## [31] "NJ" "NM" "NY" "NC" "ND" "4" "OK" "6" "PA" "RI" "SC" "SD" "TN" "5" "4"
## [46] "7" "VA" "WA" "WV" "WI" "7"
```

Soal Nomor 2

Buat fungsi `sum_n` yang dapat digunakan untuk menghitung jumlah bilangan bulat dari 1 hingga `n`. Gunakan pula fungsi ini untuk menentukan jumlah bilangan bulat dari 1 hingga 5000.

```
sum_n <- function(n) {
  1/2*(n^2+n)
}

sum_n(5000)
```

```
## [1] 12502500
```

Soal Nomor 3

Buat fungsi `compute_s_n` yang dapat digunakan untuk menghitung jumlah $S_n = 1^2 + 2^2 + 3^2 + \dots + n^2$. Tampilkan hasil penjumlahan ketika $n = 10$.

```
compute_s_n <- function(x){  
  (x*(x+1)*(2*x+1))/6  
}  
  
compute_s_n(10)
```

```
## [1] 385
```

Soal Nomor 4

Buat vektor numerik kosong dengan nama: `s_n` dengan ukuran:25 menggunakan `s_n <- vector("numeric", 25)`. Simpan di hasil komputasi S_1, S_2, \dots, S_{25} menggunakan FOR-LOOP.

```
m <- 25  
s_n <- vector("numeric", m)  
  
for(i in 1:m)  
{  
  s_n[i] <- compute_s_n(i)  
}  
  
s_n
```

```
## [1] 1 5 14 30 55 91 140 204 285 385 506 650 819 1015 1240  
## [16] 1496 1785 2109 2470 2870 3311 3795 4324 4900 5525
```

Soal Nomor 5

Ulangi langkah pada soal no.4 dan gunakan fungsi `sapply`

```
m <- 25  
s_n <- vector("numeric", m)  
  
sapply(1:25, s_n <- compute_s_n)
```

```
## [1] 1 5 14 30 55 91 140 204 285 385 506 650 819 1015 1240  
## [16] 1496 1785 2109 2470 2870 3311 3795 4324 4900 5525
```