Struktur-Kontrol-dari-R.R

Benhard Leroy

2024-09-12

# Membuat If else  
angka <- 15  
if (angka > 10) {  
 print("Angka lebih beasr dari 10")  
} else {  
 print ("Angka lebih kecil atau sama dengan 10")  
}

## [1] "Angka lebih beasr dari 10"

# Membuat Nested If  
x <- 32  
  
if (x > 15) {  
 print ("Above 15")  
 if (x > 30) {  
 print ("and also above 30")  
 } else {  
 print ("but not above 30")  
 }  
} else {  
 print ("below 15")  
}

## [1] "Above 15"  
## [1] "and also above 30"

# Membuat If and  
a <- 100  
b <- 231  
c <- 54  
  
if (b > a & c > a) {  
 print("Both conditions are true")  
}  
  
#Membuat If or  
if (b > a & c > a) {  
 print("At least one of the conditions are true")  
}  
  
# Membuat Else If  
nilai <- 85  
if (nilai >= 90) {  
 print ("Grade A")  
} else if (nilai >= 80) {  
 print ("Grade B")  
} else if (nilai >= 70) {  
 print ("Grade C")  
} else {  
 print ("Grade D")  
}

## [1] "Grade B"

# Membuat while  
count <- 4  
while (count <= 10) {  
 print (count)  
 count <- count + 2  
}

## [1] 4  
## [1] 6  
## [1] 8  
## [1] 10

# Membuat while break  
a <- 2  
while (a < 7) {  
 print (a)  
 a <- a + 4  
 if (a == 5) {  
 break  
 }  
}

## [1] 2  
## [1] 6

# Membuat while next  
b <- 3  
while (a < 4) {  
 b <- b+2  
 if (b == 5) {  
 next  
 }  
 print (b)  
}  
# Membuat If else + while loops  
  
dice <- 1  
while (dice <= 6) {  
 if (dice < 6) {  
 print ("No Yahtzee")  
 } else {  
 print ("Yahtzee!")  
 }  
 dice <- dice + 1  
}

## [1] "No Yahtzee"  
## [1] "No Yahtzee"  
## [1] "No Yahtzee"  
## [1] "No Yahtzee"  
## [1] "No Yahtzee"  
## [1] "Yahtzee!"

# Membuat for  
for (s in 1:10) {  
 print (s)  
}

## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10

# Membuat for fruits and dice  
fruits <- list("pear", "mango", "watermelon")  
  
for (x in fruits) {  
 print (x)  
}

## [1] "pear"  
## [1] "mango"  
## [1] "watermelon"

dice <- c(1,2,3,4,5,6)  
  
for (x in dice) {  
 print (x)  
}

## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6

# Membuat nested for  
adj <- list("green", "sweet", "tasty")  
  
fruits <- list("pear", "kiwi", "watermelon")  
 for (x in adj) {  
 for (y in fruits) {  
 print(paste(x,y))  
 }  
 }

## [1] "green pear"  
## [1] "green kiwi"  
## [1] "green watermelon"  
## [1] "sweet pear"  
## [1] "sweet kiwi"  
## [1] "sweet watermelon"  
## [1] "tasty pear"  
## [1] "tasty kiwi"  
## [1] "tasty watermelon"

# Membuat for break  
fruits <- list("pear", "kiwi", "watermelon")  
  
for (x in fruits) {  
 if (x == "kiwi") {  
 break  
 }  
 print(x)  
}

## [1] "pear"

# Membuat for next  
fruits <- list("pear", "kiwi", "watermelon")  
  
for (x in fruits) {  
 if (x == "watermelon") {  
 next  
 }  
 print(x)  
}

## [1] "pear"  
## [1] "kiwi"

# Membuat if else + for loops  
dice <- 1:6  
for (x in dice) {  
 if (x == 6) {  
 print(paste("The dice number is", x, "Yahtzee!"))  
 } else {  
 print(paste("The dice number is", x, "Not Yahtzee!"))  
 }  
}

## [1] "The dice number is 1 Not Yahtzee!"  
## [1] "The dice number is 2 Not Yahtzee!"  
## [1] "The dice number is 3 Not Yahtzee!"  
## [1] "The dice number is 4 Not Yahtzee!"  
## [1] "The dice number is 5 Not Yahtzee!"  
## [1] "The dice number is 6 Yahtzee!"

# Fungsi logika dalam R  
all(c(FALSE, TRUE, FALSE))

## [1] FALSE

any(c(FALSE, TRUE, FALSE))

## [1] TRUE

is.na(c(2,1,NA))

## [1] FALSE FALSE TRUE

angka <- c(4,12,24)  
hasil <- ifelse(angka > 12, "Lebih besar", "Lebih kecil atau sama")  
print(hasil)

## [1] "Lebih kecil atau sama" "Lebih kecil atau sama" "Lebih besar"

# Membuat return()  
kuadrat <- function(x) {  
 hasil <- x^3  
 return(hasil)  
 print("Ini tidak akan dicetak karena return sudah dipanggil")  
}  
  
# Membuat stop  
cekbilangan <- function(x) {  
 if (x<0) {  
 stop("Error: Bilangan negatif tidak diperbolehkan.")  
 } else {  
 print("Bilangan valid")  
 }  
}  
  
# Membuat warning  
cekNilai <- function (x) {  
 if (x == 0) {  
 warning("Peringatan: Nilai sama dengan nol")  
 }  
 print("Program tetap berjalan meskipun ada peringatan")  
}  
   
# Membuat try()  
result <- try (log(-1), silent=TRUE)

## Warning in log(-1): NaNs produced

if (inherits(result, "try\_error")) {  
 print ("Terjadi kesalahan, tidak bisa menghitung log dari angka negatif")  
} else {  
 print(result)  
}

## [1] NaN

#Membuat trycatch()  
hasil <- tryCatch({  
 log(-1)  
},  
error = function(e) {  
 print ("Terjadi kesalahan: log dari bilangan negatif tidak mungkin.")  
},  
error = function(e) {  
 print ("Peringatan: Ada yang tidak benar.")  
},  
finally = {  
 print ("Eksekusi tryCatch selesai")  
})

## Warning in log(-1): NaNs produced

## [1] "Eksekusi tryCatch selesai"