

1. Deklarasi

Integer : nilai_uang

Integer : jumlah_pecahan_rp1000, jumlah_pecahan_rp500

Integer : jumlah_pecahan_rp100, jumlah_pecahan_rp50

Integer : jumlah_pecahan_rp25

Deklarasi

Write (Masukkan nilai uang dalam kelipatan 25")

Read (nilai_uang)

Semua jumlah_pecahan <- 0

While nilai_uang >0 do:

 If nilai_uang >= 1000 then :

 Tambahkan 1 ke jumlah_pecahan_rp1000

 Kurangi nilai_uang

 Else if nilai_uang >= 500 then :

 Tambahkan 1 ke jumlah_pecahan_rp500

 Kurangi nilai_uang dengan 500

 Else if nilai_uang >= 100 then :

 Tambahkan 1 ke jumlah_pecahan_rp100

 Kurangi nilai_uang dengan 100

 Else if nilai_uang ke jumlah >= 50 them :

 Tambahkan 1 ke jumlah_pecahan_rp50

 Kurangi nilai_uang dengan 50

 Else if nilai_uang >= 25 then :

 Tambahkan 1 ke jumlah_pecahan_rp25

 Kurangi nilai uang_dengan 25

Write (" Jumlah Pecahan Rp.1000 : " , jumlah_Pecahan_rp1000

Write (" Jumlah Pecahan Rp.500 : " , jumlah_Pecahan_rp500

Write (" Jumlah Pecahan Rp.100 : " , jumlah_Pecahan_rp10

Write (" Jumlah Pecahan Rp.50 : " , jumlah_Pecahan_rp50

Write (" Jumlah Pecahan Rp.25 : " , jumlah_Pecahan_rp25

2. Deklarasi

Kantung_permen : integer

Deskripsi

Write (" masukkan jumlah permen dalam kantung :")

Read (kantung_permen)

While kantung_permen > 0 do :

 If kantung_permen >-3 then :

 Berikan satu permen kepada anak pertama

 Berikan satu permen pada anak kedua

 Berikan satu permen pada anak ketiga

 Kurangi kalung_permen dengan 3

Else :

Write (“ tidak cukup permen untuk semua anak “)

Keluar lop

Write (“ setiap anak telah menerima permen yang sama”)

Pseudecode itu memperbaiki kesalahan dengan menambahkan pengecekan if untuk memastikan bahwa kalung permen tidak kosong sebelum memberikan peremen kepada anak-anak.