**Program** FrekuensiKataPertama

**KAMUS LOKAL**

USE MesinKata1

kata : Kata

banyakkata, frekuensi, i : integer

sama : Boolean

**ALGORITMA**

banyakkata <- 0

frekuensi <- 0

startKata

if (endKata) then

output (“Pita karakter kosong.”)

else

kata1.buffer <- currentKata.buffer

katapertama.length <- currentKata.length

frekuensi <- frekuensi + 1

banyakkata <- banyakkata + 1

while (not (endKata)) do

advKata

if (not(endKata)) then

sama <- true

if (katapertama.length != currentKata.length) then

sama <- false

else

i traversal [0..{katapertama.length)-1]

if (currentKata.buffer[i] != katapertama.buffer[i]) then

Sama <- false

if (sama) then

frekuensi <- frekuensi + 1

banyakkata <- banyakkata + 1

output (frekuensi, “/”, banyakkata)

**Program** Anagram

**KAMUS**

USE MesinKata1

function isAnagram (kata1 : Kata, k2 : Kata) -> boolean

katapertama : Kata

countanagram : integer

**function** isAnagram(kata1 : Kata, k2 : Kata) -> boolean

**KAMUS LOKAL**

count = array[0..N\_MAX – 1] of integer

i, j : integer

found : boolean

**ALGORITMA LOKAL**

if (kata1.length != kata2.length) then

->false

else

i traversal [0..N\_MAX-1]

count[i] <- 0

i traversal [0..kata1.length -1 ]

found <- false

j traversal [0..kata1.length – 1]

if ((kata1.buffer[i] == kata2.buffer [j]) and (count [j] = 0)) then

count[j] <- count[j] + 1

found <- true

break

if (not(found)) then

->false

->true

**ALGORITMA**

Countanagram <- 0

startKata

if (endKata) then

output(“Pita karakter kosong”)

else

katapertama <- currentKata

advKata

while (not(endKata)) do

if (isAnagram(katapertama,currentKata)) then

countanagram <- countanagram + 1

advKata

output(“Banyaknya anagram dengan kata pertama : “, countanagram)

**Program** MesinToken

**KAMUS**

USE MesinKata1

arr\_kata = array [0..N\_MAX-1] of Kata

i : integer

**function** IsOperan (token : Kata) -> boolean

**function** Infix (operan1 : Kata, operan2 : Kata, operator : Kata) -> Kata

**function** isOperan (token : Kata) -> Boolean

**KAMUS LOKAL**

**ALGORITMA**

->((token.buffer[0] = "^") or (token.buffer[0] = "-") or (token.buffer[0] = "+") or (token.buffer[0] = "/") or (token.buffer[0] = "\*"))

**function** Infix (operan1 : Kata, operan2 : Kata, operator : Kata) -> Kata

**KAMUS LOKAL**

i, j : integer

result : Kata

**ALGORITMA**

i <- 0

result.buffer[i] = “(“

i <- i + 1

j traversal [0..(operan1.length-1)]

result.buffer[i] <- operan1.buffer[j]

i <- i + 1

result.buffer[i] <- BLANK

i <- i + 1

result.buffer[i] <- operator.buffer[0]

i <- i + 1

result.buffer[i] <- BLANK

j traversal [0..(operan2.length-1)]

result.buffer[i] <- operan2.buffer[j]

i <- i + 1

result.buffer[i] <- “)”

i <- i + 1

result.length <- i

->result

**ALGORITMA UTAMA**

i <- 0

startKata

if (endKata) then

output (“Pita karakter kosong.”)

else

while (not(endKata)) do

if (not(isOP(currentKata)) then

arr\_kata[i].buffer <- currentKata.buffer

arr\_kata[i].length <- currentKata.length

i <- i + 1

else

arr\_kata[i-2]<- Infix (arr\_kata[i-2], arr\_kata[i-1], currentKata)

i <- i – 1

adv

output(arr\_kata[0])