

Rania Sasi Kirana

18221168

LATIHAN 1

Program FrekuensiKataPertama

{ Menghitung frekuensi kemunculan kata pertama dalam pita. }

KAMUS

{ *** Mesin yang digunakan *** }

USE MesinKata1

nbKata: integer { banyaknya kata dalam pita }

nKataPertama: integer { frekuensi kemunculan kata pertama dalam pita }

i: integer

cek: boolean

kataPertama: Kata

ALGORITMA

nbKata <- 0

nKataPertama <- 0

startKata

if endKata then

 output("Pita karakter kosong")

else

 { Menyimpan informasi kata pertama }

 kataPertama.length <- currentKata.length

 i traversal [0..(kataPertama.length)-1]

 kataPertama.buffer[i] <- currentKata.buffer[i]

 advKata

```

while not endKata do

  cek <- True

  if (kataPertama.length != currentKata.length) then
    cek <- False
  else
    i traversal [0..(kataPertama.length)-1]
    if (kataPertama.buffer[i] != currentKata.buffer[i]) then
      cek <- False

  if cek then
    nKataPertama <- nKataPertama + 1

  nbKata <- nbKata + 1
  advKata

  output("Frekuensi kata pertama dalam pita adalah", nKataPertama, "/", nbKata)

```

LATIHAN 2

```

function isKataAnagram (k1,k2: Kata) -> boolean
{ Menghasilkan True jika k1 dan k2 adalah anagram }

```

KAMUS LOKAL

```
{ *** Mesin yang digunakan *** }
```

```
USE MesinKata1
```

```
i,j: integer
```

```
bener: integer { berfungsi sebagai bool untuk mengetahui apakah semua elemen dalam k1 sama dengan k2 }
```

```
cek: boolean
```

ALGORITMA LOKAL

```
if (K1.length == K2.length) then
  bener <- 0
  i traversal [0..K1.length[i]-1] { cek setiap elemen dalam k1 }
  cek <- True
  while cek do
    j traversal [0..K2.length[i]-1] { cek setiap elemen dalam k2 apakah ada yg sama dengan k1 }
    if (k1.buffer[i] == k2.buffer[j]) then
      k2.buffer[j] <- 0 { delete elemen yang sama dengan k1 kemudian lanjut cek elemen lain }
      bener <- bener + 1
      cek <- False

  if (bener == K1.length) then { elemen dalam k1 dan k2 semua sama }
    -> True
  else
    -> False
```

Program hitungAnagram

{ Menghitung ada berapa banyak kata yang anagram dengan kata pertama pada pita karakter (tidak termasuk kata pertama)

Kata dikatakan anagram jika memiliki panjang yang sama, terdiri atas huruf yang sama dan masing-masing huruf memiliki jumlah yang sama. }

KAMUS

{ *** Mesin yang digunakan *** }

USE MesinKata1

nAnagram: integer { menghitung banyak anagram }

i: integer

```
function isKataAnagram (k1, k2: Kata) -> boolean  
{ Menghasilkan True jika k1 dan k2 adalah anagram }
```

ALGORITMA

```
nAnagram <- 0  
startKata  
if endKata then  
    output("Pita karakter kosong")  
else  
    { Menyimpan informasi kata pertama }  
    kataPertama.length <- currentKata.length  
    i traversal [0..(kataPertama.length)-1]  
        kataPertama.buffer[i] <- currentKata.buffer[i]  
    advKata  
    while not endKata do  
        if (isKataAnagram(kataPertama, currentKata)) then  
            nAnagram <- nAnagram + 1  
        advKata  
  
    output("Banyak kata yang anagram dengan kata pertama sebanyak ", nAnagram)
```

LATIHAN 3

Program mesinToken

KAMUS

```
{ *** Mesin yang digunakan *** }
```

USE MesinKata1

ALGORITMA

startKata

if endKata then

 output("Pita karakter kosong")

else

 angka1 <- currentKata

 advKata

 angka2 <- currentKata

 advKata

 operator1 <- currentKata

 advKata

 angka3 <- currentKata

 advKata

 angka4 <- currentKata

 advKata

 operator2 <- currentKata

 advKata

 operator3 <- currentKata

 output("(" , angka1, operator1, angka2, ")", operator2, "(" , angka3, operator3, angka4, ")")