UTS ALPRO 2022

"JAWABAN HANYA REFERENSI"

No.1

Potongan Teks Algoritma

Loop ke -1 : I = 1 , j = 0

Loop ke -2 : I = 2, j = 4

Loop ke -3 : I = 3, j = 40

Loop ke -4 : I =4 , j = 44

Loop ke -5 = I = 5 , j = 1144

Jadi nilai akhir dari J pada eksekusi tersebut yaitu 1144.

No.2

```
Program PenurunanSdanP

{menampilkan banyak iterasi yang dibutuhkan untuk menurunkan S atau P menjadi nol atau negatif}


Kamus

    S : integer {variabel masukan}

    P : integer {variabel masukan}

  nIterasi :integer{banyak iterasi untuk menurunkan patau s menjadi nol atau negatif}


Algoritma

    input(S,P)

    if (S <= 0)OR(P <= 0) then

        output("S dan P harus positif")

    else

       if (P>S) then
```

```
            output("P harus kurang dari sama dengan S")
      else
          nIterasi ← 0
          while (S > 0) AND (P > 0) do
              if ((P MOD 2)=0) then
                   S ← S-1
              else {(P MOD 2)!=0} {ganjil}
                   S ← S-2
              P ← P – 2
               nIterasi ← nIterasi + 1
             output(nIterasi)
```

No.3

```
     SplitArray(T,T1,T2)

Procedure SplitArr(input: T: array [1..N] of integer,output: T1,T2: array
[1..N] of integer) {Membagi array T menjadi 2 array baru, yaitu T1 dan T2}
{aturan split, jika panjang array T genap maka banyaknya elemen T1 dan T2
adalah sama. Jika panjang array T ganjil, maka elemen T1 akan lebih banyak 1
elemen dibandingkan T2} {Contoh: T= 1 2 2 1 --- menghasilkan T1= 1 2 dan T2=
2 1} {Contoh: T= 1 2 3 2 1 --- menghasilkan T1= 1 2 3 dan T2= 2 1} {IS: T
sembarang} {FS: menghasilkan array baru T1 dan T2} {Kamus Lokal} {Algoritma}


Kamus Lokal

    N : integer {banyaknya elemen dalam array}
    i : integer {counter}
   split : integer {variabel jumlah N setelah dibagi 2}
   T : array [1..N] of integer {deklarasi array integer}


Algoritma

  Panjang ← PanjangArr(T)
  Split ← (Panjang DIV 2)
  if(Panjang MOD 2 != 0) then
     I traversal [1..Panjang]
     If(I <= Split+1)then
```

```
                T1i ← Ti
     else
            b = I – (Split+1)
             T2b ←Ti
  Else
     I traversal [1..Panjang]
     If(I <= Split)then
              T1i ←Ti
     else
              b = I – (Split)
              T2b ←Ti
→ T1
→ T2
```

No.4

```
Functions IsPolindron (T: array [1..N] of integer) → boolean {Mengirimkan
true jika elemen array membentuk Polindron, dan false jika tidak membentuk
polindron. Elemen table T polindron jika elemen pertama sama dengan yang
terakhir, elemen kedua sama dengan elemen terakhir - 1, elemen ketiga sama
dengan elemen terakhir – 2, … dll}

{Contoh: T= 1 2 3 2 1 --- true; T= 1 2 3 4 5 --- false; T= 3 4 5 6 6 5 4 3 -
-- true}

Kamus Lokal

     lenArr : integer {panjang array T}

     tengah : integer {indeks tengah dari T}

     asumsi : boolean {asumsi pertama array T adalah Polindron}

Algoritma

     lenArr ← PanjangArr(T)

     asumsi ← True

     if ((lenArr MOD 2)=0) then {genap}

                 tengah ← lenArr / 2

     else {(lenArr MOD 2)!=0} {ganjil}

          tengah ← (lenArr / 2) +1
```

```
    i traversal [1..tengah]
        if (Ti != TlenArr-i) then
            asumsi ← False
    → asumsi
```