

## BAB 14

1. Pada deklarasi yang pertama, dibuat record yang memiliki 2 buah field bertipe real. Lalu dibuat matriks dengan tipe data record tersebut. Jika dibuat dalam code c, hasilnya:

```
const NBarisMaks = 5;

const NKolomMaks = 5;

struct TP{

    float T;

    float P;

};

struct TP B[NBarisMaks][NKolomMaks];
```

sehingga terdapat 25 buah indeks dalam array B. Indeks matriks dapat diakses dengan:

```
B[0][0].T;
```

atau

```
B[0][0].P;
```

Sedangkan pada deklarasi yang kedua, dibuat record yang memiliki 2 buah field dengan matriks yang bertipe data float. Lalu dibuat variabel dengan tipe data struct tersebut. Jika dibuat dalam code C, hasilnya:

```
const NBarisMaks = 5;

const NKolomMaks = 5;

struct TP{

    float T[NBarisMaks][NKolomMaks];

    float P[NBarisMaks][NKolomMaks];

};

struct TP B;
```

Indeks matriks dapat diakses dengan:

```
B.T[0][0];
```

atau

```
B.P[0][0];
```

6. Procedure JumlahNilai(input m, n : integer, NR : array [1..m] of real, NilaiMhs : array [1..m, 1..n] of Nilai,

output NR : array [1..m] of real)

{ prosedur untuk memasukkan dan menjumlahkan nilai dan sks mahasiswa }

{ K.Awal: nilai m dan n terdefinisi isinya }

{ K.Akhir: larik NR dan NilaiMhs terdefinisi isinya }

#### DEKLARASI

sigma1 : real { menyimpan nilai total sigma di pembilang }

sigma2 : real { menyimpan nilai total sigma di penyebut }

#### ALGORITMA

##### START

sigma1 <- 0

sigma2 <- 0

{ looping untuk menginput nilai dan sks mahasiswa }

for i <- 0 to m do

write("Data Mahasiswa i+1")

for j <- 0 to n do

ULANG:

{ input indeks nilai dan SKS mata kuliah }

write("Indeks Nilai Mata Kuliah j+1: ")

read(NilaiMhs[i][j].indeksNilai)

case NilaiMhs[i][j].indeksNilai

'A','a': NilaiMhs[i][j].indeksNilai <- 4

'B','b': NilaiMhs[i][j].indeksNilai <- 3

'C','c': NilaiMhs[i][j].indeksNilai <- 2

'D','d': NilaiMhs[i][j].indeksNilai <- 1

'E','e': NilaiMhs[i][j].indeksNilai <- 0

otherwise: goto ULANG { kembali ke read(NilaiMhs[i][j].indeksNilai) }

write("SKS Mata Kuliah j+1: ")

read(NilaiMhs[i][j].SKS)

{ menjumlahkan nilai sigma1 dan sigma2 }

sigma1 <- sigma1 + (NilaiMhs[i][j].indeksNilai \* NilaiMhs[i][j].SKS)

sigma2 <- sigma2 + NilaiMhs[i][j].SKS

endfor

{ menjumlahkan nilai rata-rata }

NR[i] <- sigma1 / sigma2

endfor

END

program HitungNilaiRerata

{ program untuk menghitung nilai rata-rata mahasiswa }

DEKLARASI

m : integer { jumlah mahasiswa }

n : integer { jumlah mata kuliah }

Nilai : record < SKS : integer, indeksNilai : integer >

NilaiMhs : array [1..m, 1..n] of Nilai

NR : array [1..m] of real

procedure JumlahNilai(input m, n : integer, NR : array [1..m] of real, NilaiMhs : array [1..m, 1..n] of Nilai,

output NR : array [1..m] of real)

{ prosedur untuk memasukkan dan menjumlahkan nilai dan sks mahasiswa }

ALGORITMA

START

write("Jumlah Mahasiswa: ")

read(m)

write("Jumlah Mata Kuliah: ")

read(n)

JumlahNilai(NR, NilaiMhs, m, n)

{ looping unuk mencetak nilai rata-rata tiap mahasiswa }

for i <- 0 to m do

write("Nilai rata-rata Mahasiswa i+1: NR[i]\n")

endfor

END

## BAB 15

1. Procedure CariNilai(input A : array [1..n] of integer, n : integer, x : integer,

output id : integer)

{ Prosedur untuk mencari nilai x di dalam larik A dan mengisi id dengan index ditemukannya nilai x. Jika tidak ditemukan, maka id diisi dengan -1 }

{ K.Awal: larik A, n, dan x terdefinisi nilainya }

{ K.Akhir: id terisi nilainya }

### DEKLARASI

i : integer { pencacah pengulangan }

### ALGORITMA

START

id <- -1

for i <- n-1 downto 0 do

if A[i] = x then

id <- i { mengisi nilai id dengan indeks i }

break

endif

endfor

END

### Program PencarianBeruntunDariAkhir

{ Program pencarian beruntun yang memberikan hasil indeks elemen larik yang mengandung x, tetapi pencarian dimulai dari elemen terakhir }

## DEKLARASI

n : integer { banyaknya elemen larik }

A : array [1..n] of integer

x : integer { nilai yang akan dicari indeks-nya }

id : integer { nilai indeks yang akan dicetak }

procedure CariNilai(input A : array [1..n] of integer, n : integer, x : integer,  
output id : integer)

{ Prosedur untuk mencari nilai x di dalam larik A dan mengisi id dengan index ditemukannya nilai x. Jika tidak ditemukan, maka id diisi dengan -1 }

## ALGORITMA

### START

{ memasukkan nilai banyaknya elemen }

read(n)

for i <- 0 to n do

read(A[i])

endfor

{ memasukkan nilai yang ingin dicari }

read(x)

CariNilai(A, n, x, id)

{ mencetak hasil akhir jika ditemukan atau tidak }

if id != -1 then

write("Nilai x berada di index ke-", x, id)

else

```

    write("Nilai x tidak ditemukan!")
endif
END

```

5. Procedure CariNilai(input A : array [1..n] of integer, x,n : integer,

output ketemu : boolean, id : integer)

{ Prosedur untuk mencari nilai x di larik lalu mengubah nilai ketemu menjadi true jika didapat nilai x dalam larik serta nilai id terisi dengan k jika ketemu }

{ K.Awal: larik A, x, dan n terdefinisi nilainya }

{ K.Akhir: ketemu menjadi true atau tetap false dan id diisi dengan nilai k }

#### DEKLARASI

i : integer { peubah untuk menentukan index awal pencarian }

j : integer { peubah untuk menentukan index akhir pencarian }

k : integer { peubah untuk menentukan index tengah pencarian }

#### ALGORITMA

##### START

i <- 0

j <- n-1

while (not ketemu) and (i <= j) do

k <- i + (j - i) \* (x - A[i]) div (A[j] - A[i])

if A[k] = x then

ketemu <- true

id <- k

```

else
    if A[k] > x then
        i <- k + 1
    else
        j <- k - 1
    endif
endif
endwhile
END

```

program PencarianInterpolasi

{ Program untuk mencari nilai x di suatu larik dengan metode interpolasi. Nilai n, larik A, dan x telah terdefinisi }

#### DEKLARASI

```

n : integer { banyaknya elemen larik }
x : integer { nilai yang dicari }
A : array [1..n] of integer
ketemu : boolean
i : integer { peubah untuk menentukan index awal pencarian }
id : integer { index dimana nilai x ditemukan }

```

#### ALGORITMA

##### START

```

ketemu <- false

```

read(n)

for i <- 0 to n do

    read(A[n])

endfor

read(x)

CariNilai(A, x, n, ketemu)

if ketemu then

    write("Nilai x terdapat di index ke-", x, id)

else

    write("Nilai x tidak terdapat di larik!", x)

endif

END