

ALGORITMA & STRUKTUR DATA

Nama : Andi Farhan Sappewali

Nim : D121211078

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 kode C, hasilnya:

```
const NbarisMaks = 5;
const MkolomMaks = 5;
struct TP{
    float T;
    float P;
};
struct TP B[NbarisMaks][MkolomMaks];
```

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 bertipe data float. Lalu dibuat variabel dengan tipe data struct tersebut. Jika dibuat dalam kode 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 : int, NR : array[1..m] of real, NilaiMhs[1..m, 1..n] of Nilai, output NR : array [1..m] of real)

{prosedur untuk memasukkan dan menjumlahkan nilai 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

sigma1 \leftarrow 0

sigma2 \leftarrow 0

{looping untuk menginput nilai dan sks mahasiswa}

for i \leftarrow 0 to m do

 write("Data Mahasiswa i+1")

 for j \leftarrow 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 \leftarrow 4

 'B' 'b' : NilaiMhs[i][j].indeksNilai \leftarrow 3

 'C' 'c' : NilaiMhs[i][j].indeksNilai \leftarrow 2

 'D' 'd' : NilaiMhs[i][j].indeksNilai \leftarrow 1

 'E' 'e' : NilaiMhs[i][j].indeksNilai \leftarrow 0

 otherwise : goto ulang

```

        write("SKS Mata Kuliah j+1 : ")
        read(NilaiMhs[i][j].SKS)
        { menjumlahkan nilai sigma1 dan sigma 2 }
        sigma1  $\leftarrow$  sigma1 + (NilaiMhs[i][j].indeksNilai * NilaiMhs[i][j].SKS)
        sigma2  $\leftarrow$  sigma2 + NilaiMhs[i][j].SKS
    endfor
    { menjumlahkan nilai rata-rata }
    NR[i]  $\leftarrow$  sigma1 / sigma 2
endfor
END

```

PROGRAM HitungNilaiRata-rata

{program untuk menghitung nilai rata-rata mahasiswa}

DEKLARASI

```

m : int {jumlah mahasiswa}
n : int {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 mahasiwa}

```

ALGORITMA

```

write("Jumlah Mahasiswa : ")
read(m)

write("Jumlah Mata Kuliah: ")
read(n)

```

JumlahNilai(NR, NilaiMhs, m,n)

{looping untuk mencetak nilai rata-rata tiap mahasiswa}

for i \leftarrow 0 to m do

write(“Nilai rata-rata Mahasiswa i+r : NR[i]\n”)

endfor

END

Hasil Program

```
C:\WINDOWS\system32\cmd.exe

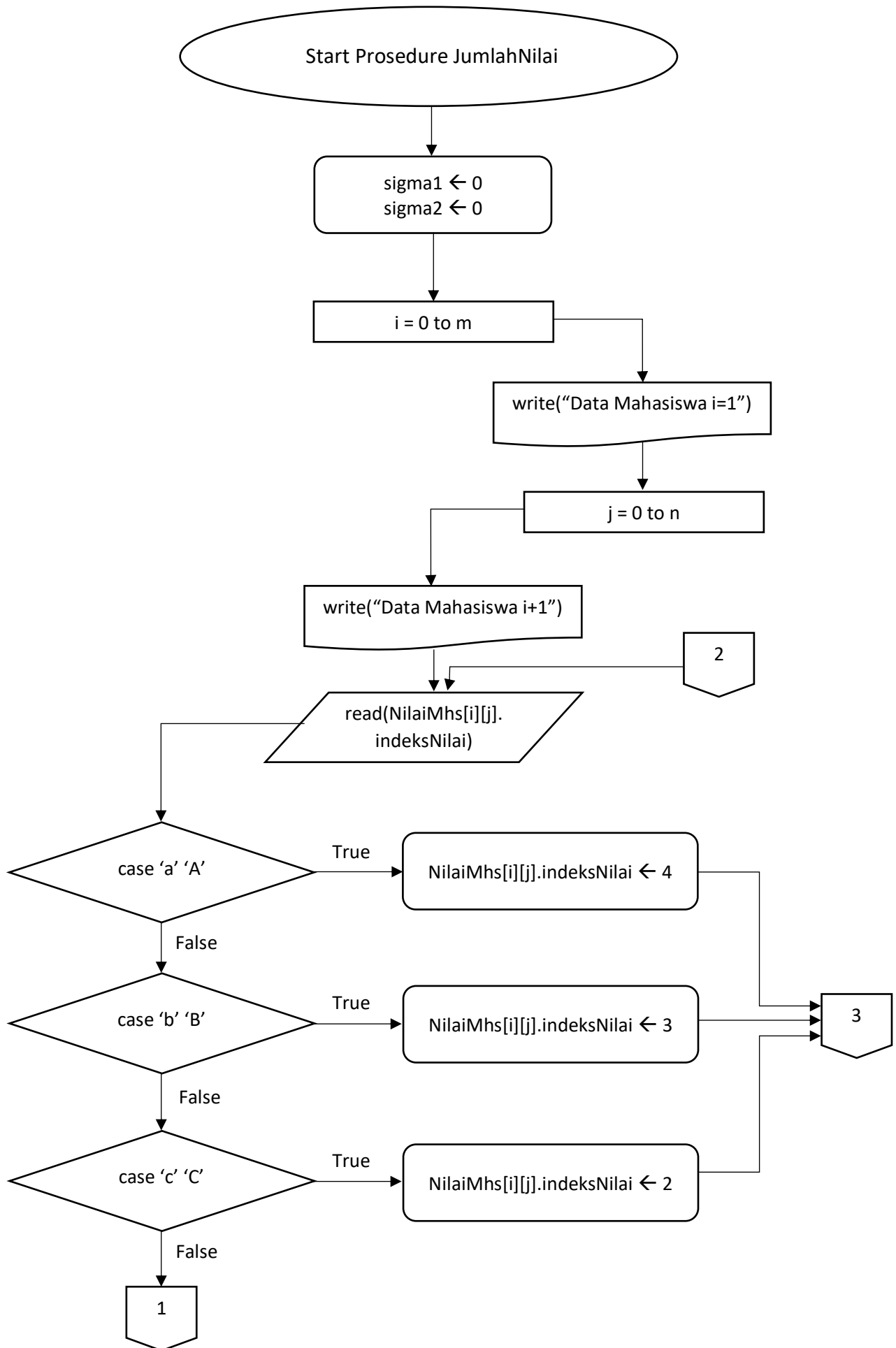
D:\E\TI\MK\SEMESTER 2\ASD>D121211078_T7_146
Jumlah Mahasiswa: 2
Jumlah Mata Kuliah: 4

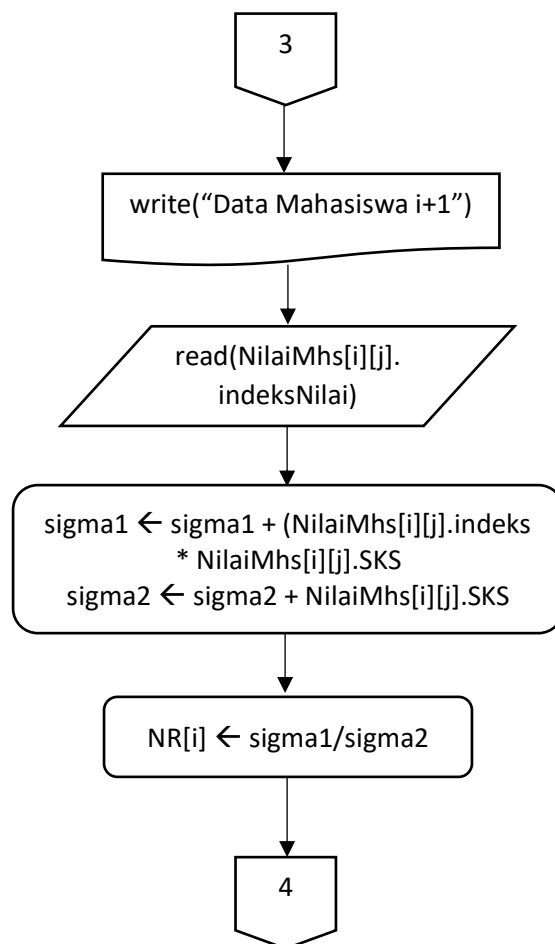
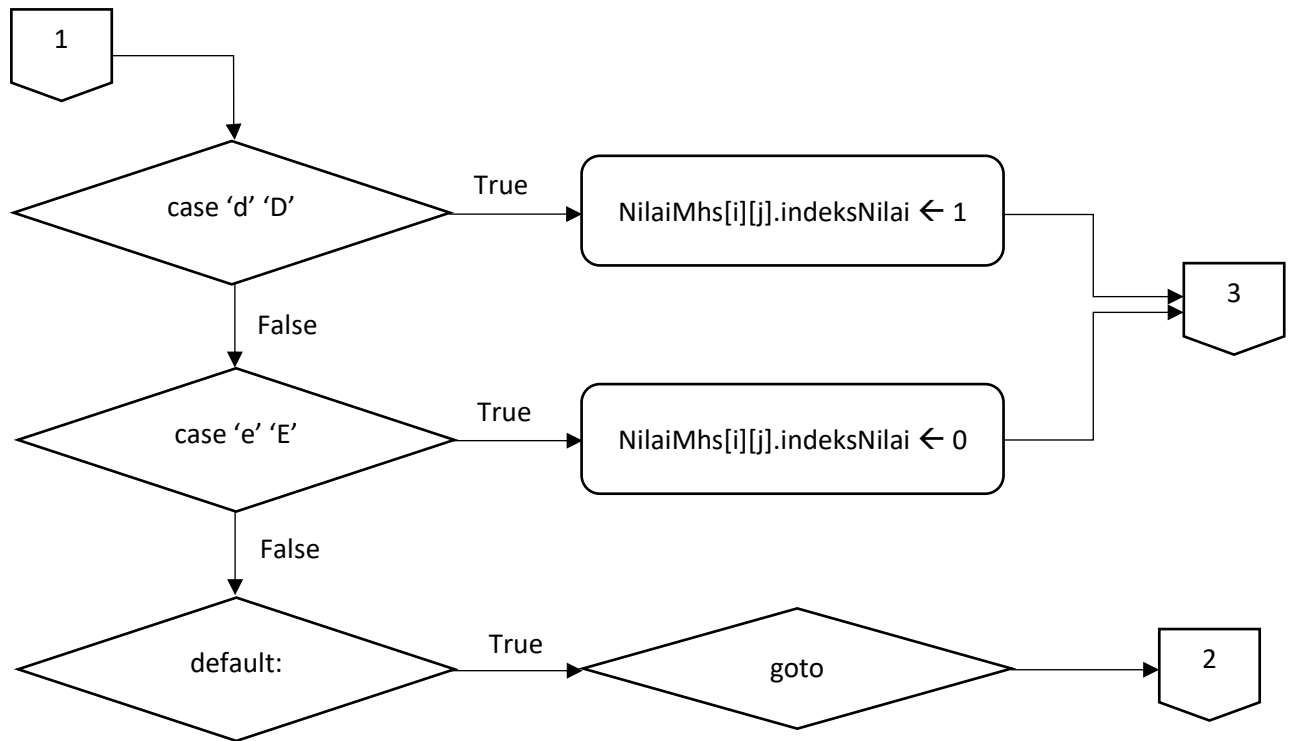
Data Mahasiswa 1
Indeks Nilai Mata Kuliah 1 (A/B/C/D/E): a
SKS Mata Kuliah 1: 4
Indeks Nilai Mata Kuliah 2 (A/B/C/D/E): b
SKS Mata Kuliah 2: 2
Indeks Nilai Mata Kuliah 3 (A/B/C/D/E): b
SKS Mata Kuliah 3: 3
Indeks Nilai Mata Kuliah 4 (A/B/C/D/E): a
SKS Mata Kuliah 4: 3

Data Mahasiswa 2
Indeks Nilai Mata Kuliah 1 (A/B/C/D/E): a
SKS Mata Kuliah 1: 4
Indeks Nilai Mata Kuliah 2 (A/B/C/D/E): a
SKS Mata Kuliah 2: 2
Indeks Nilai Mata Kuliah 3 (A/B/C/D/E): a
SKS Mata Kuliah 3: 3
Indeks Nilai Mata Kuliah 4 (A/B/C/D/E): b
SKS Mata Kuliah 4: 3

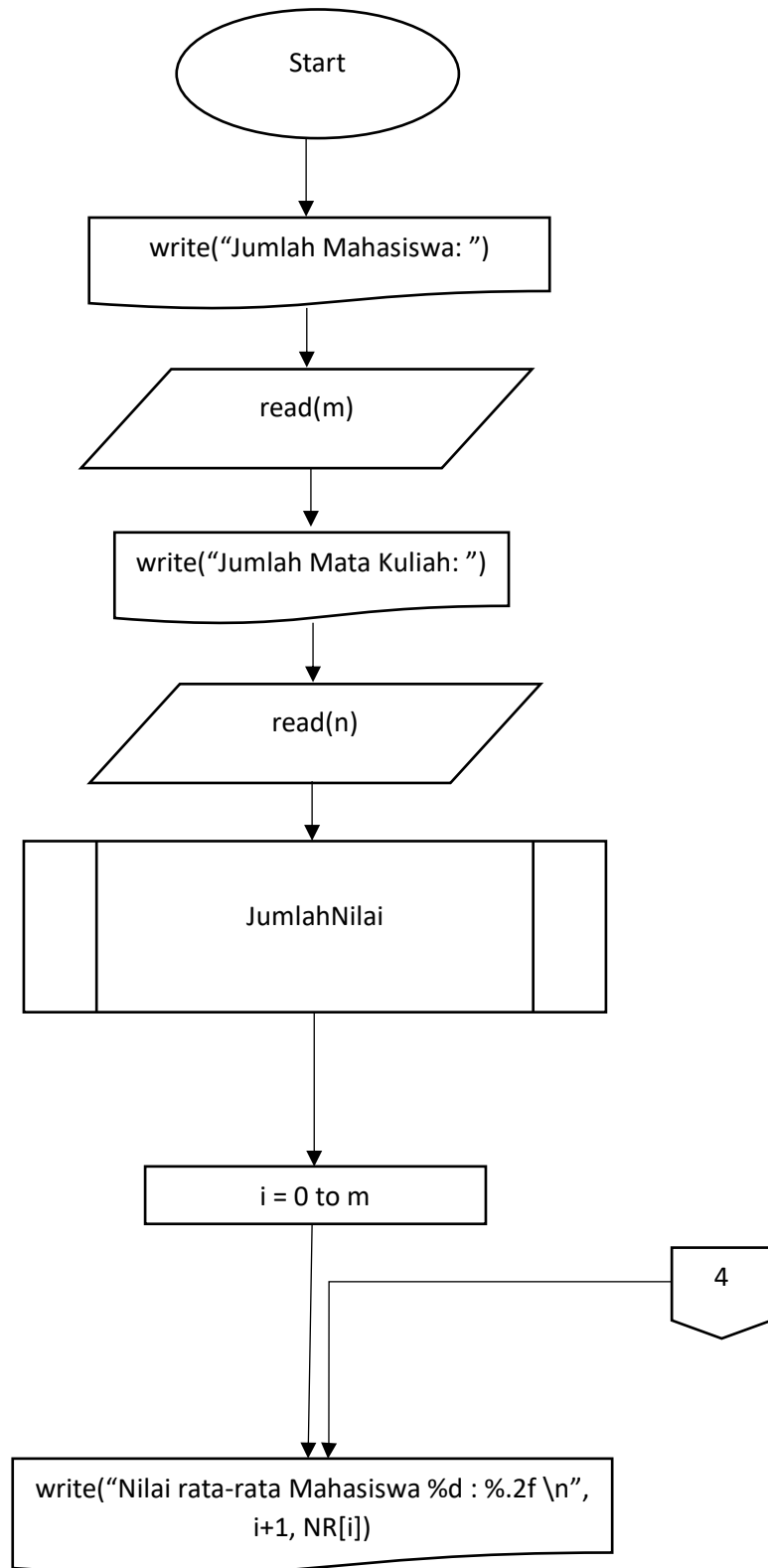
Nilai rata-rata Mahasiswa 1: 3.58
Nilai rata-rata Mahasiswa 2: 3.67
```

FLOWCHART





Prosedur JumlahNilai



Bab 15

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

{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.Awal : id terisi nilainya}

DEKLARASI

i \leftarrow integer

ALGORITMA

id \leftarrow 1

for i \leftarrow n-1 down to 0 do

 if A[i] = x then

 id \leftarrow n

 break

 endif

endfor

END

PROGRAM PencarianBeruntunDariAkhir

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

DEKLARASI

n : integer

A : array [1..n] of integer

x : integer {nilai yang akan dicari indeksnya}

id : integer {nilai yang akan menjadi output}

ALGORITMA

```
read(n)
for i ← 0 to n do
    read (A[i])
endfor

read(x)

CariNilai (A, n, x, id)

{mencetak hasil akhir}
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 benar dan nilai id terisi dengan k jika didapat nilai x dalam larik}

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

{K.Akhir : ketemu menjadi benar atau salah dan id diisi dengan nilai k}

DEKLARASI

i : integer {peubah untuk menentukan indeks awal pencarian}

j : integer {peubah untuk menentukan indeks akhir pencarian}

k : integer {peubah untuk menentukan indeks tengah pencarian}

ALGORITMA

```
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 ← benar
        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

ketemu \leftarrow salah

read(n)

for i \leftarrow 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 terdapat di larik!", x)

endif

END

Hasil Program

```
C:\WINDOWS\system32\cmd.exe

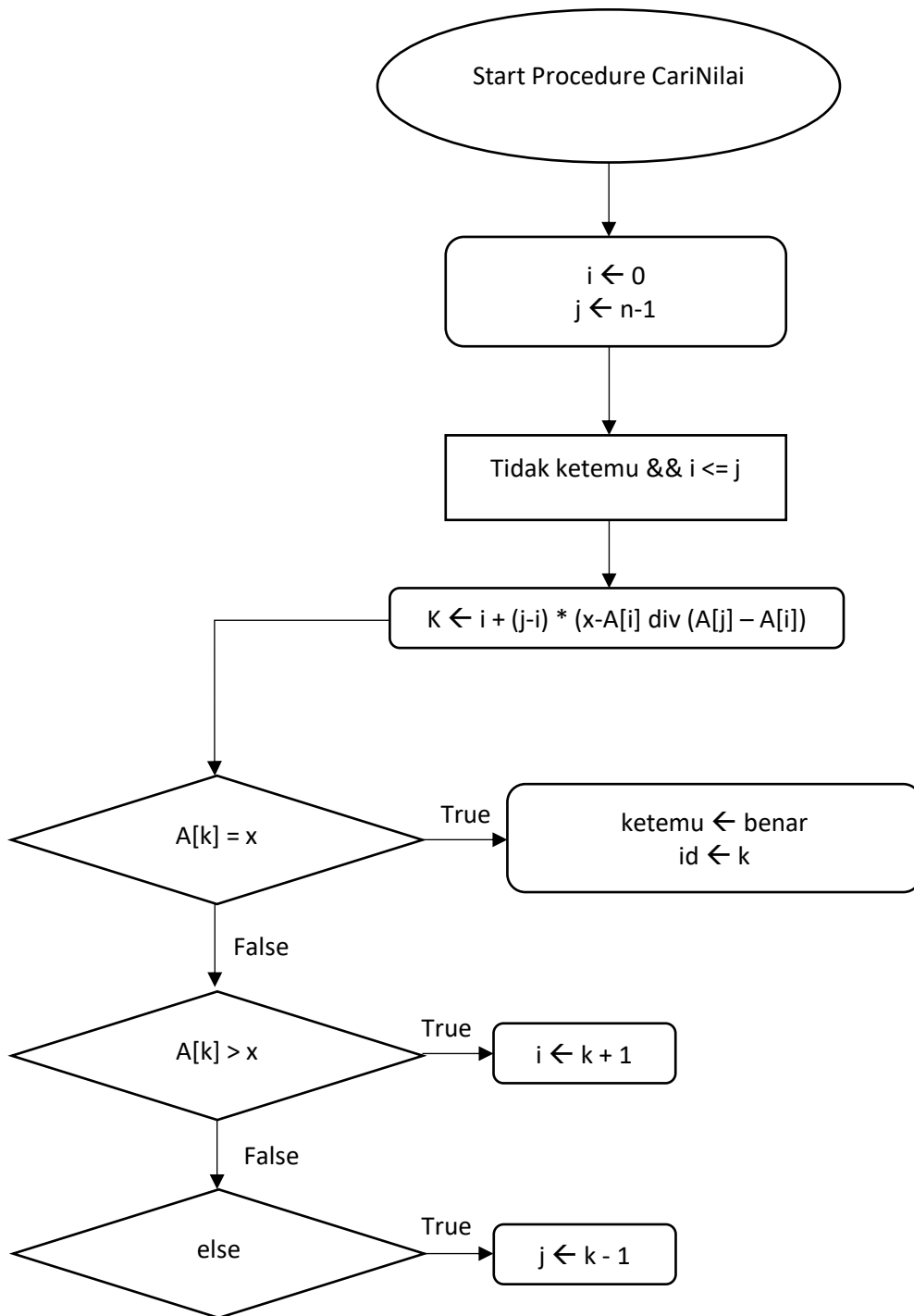
D:\E\TI\MK\SEMESTER 2\ASD>D121211078_T7_155
Tentukan banyak elemen: 4

Masukkan nilai dengan urutan dari terkecil ke terbesar
Nilai A[0] : 13
Nilai A[1] : 15
Nilai A[2] : 16
Nilai A[3] : 17
Nilai yang dicari dalam larik: 15
Nilai 15 terdapat di index ke-1

D:\E\TI\MK\SEMESTER 2\ASD>D121211078_T7_155
Tentukan banyak elemen: 5

Masukkan nilai dengan urutan dari terkecil ke terbesar
Nilai A[0] : 13
Nilai A[1] : 15
Nilai A[2] : 17
Nilai A[3] : 18
Nilai A[4] : 19
Nilai yang dicari dalam larik: 16
Nilai 16 tidak terdapat di larik!
```

FLOWCHART



Procedure CariNilai

