

# ALGORITMA PEMROGRAMAN

## Pertemuan X

# ARRAY MULTI DIMENSI

Oleh  
**Achmad Arrosyidi**



# TUJUAN PEMBELAJARAN

## Umum:

- ✓ Mahasiswa dapat merekayasa Array multi dimensi dalam bentuk flowchart.

## Khusus:

- ✓ Mahasiswa dapat menerapkan Array multi dimensi dalam bentuk flowchart.



# MATERI KULIAH

1. Konsep Array
2. Karakteristik Jenis Array Multi Dimensi
3. Deklarasi dan Inisialisasi Array Multi Dimensi
4. Pengisian / Assignment Array Multi Dimensi
5. Pengoperasian Array Multi Dimensi
6. Contoh Penerapan Array Multi Dimensi
7. Latihan



# 1. KONSEP ARRAY



# 1. KONSEP ARRAY MULTI DIMENSI

- Array adalah suatu alokasi beberapa tempat di memori yang tersimpan secara berurutan yang digunakan untuk menyimpan beberapa nilai dengan tipe data yang homogen.
- Ukuran atau jumlah elemen maksimum array telah diketahui dari awal yaitu ketika array dibuat.
- Sekali ukuran array ditentukan maka tidak dapat diubah.
- Ukuran array adalah bilangan bulat positif.
- Array harus diberi nama sebagai identifikasi.
- Cara mengaksesnya adalah dengan menyebutkan nama array dan indeksinya.
- Indeks array dimulai dari 0 sampai dengan  $n-1$  ( $n$  adalah ukuran array).



## **2. KARAKTERISTIK ARRAY MULTI DIMENSI**



## 2.1. KARAKTERISTIK ARRAY MULTI DIMENSI

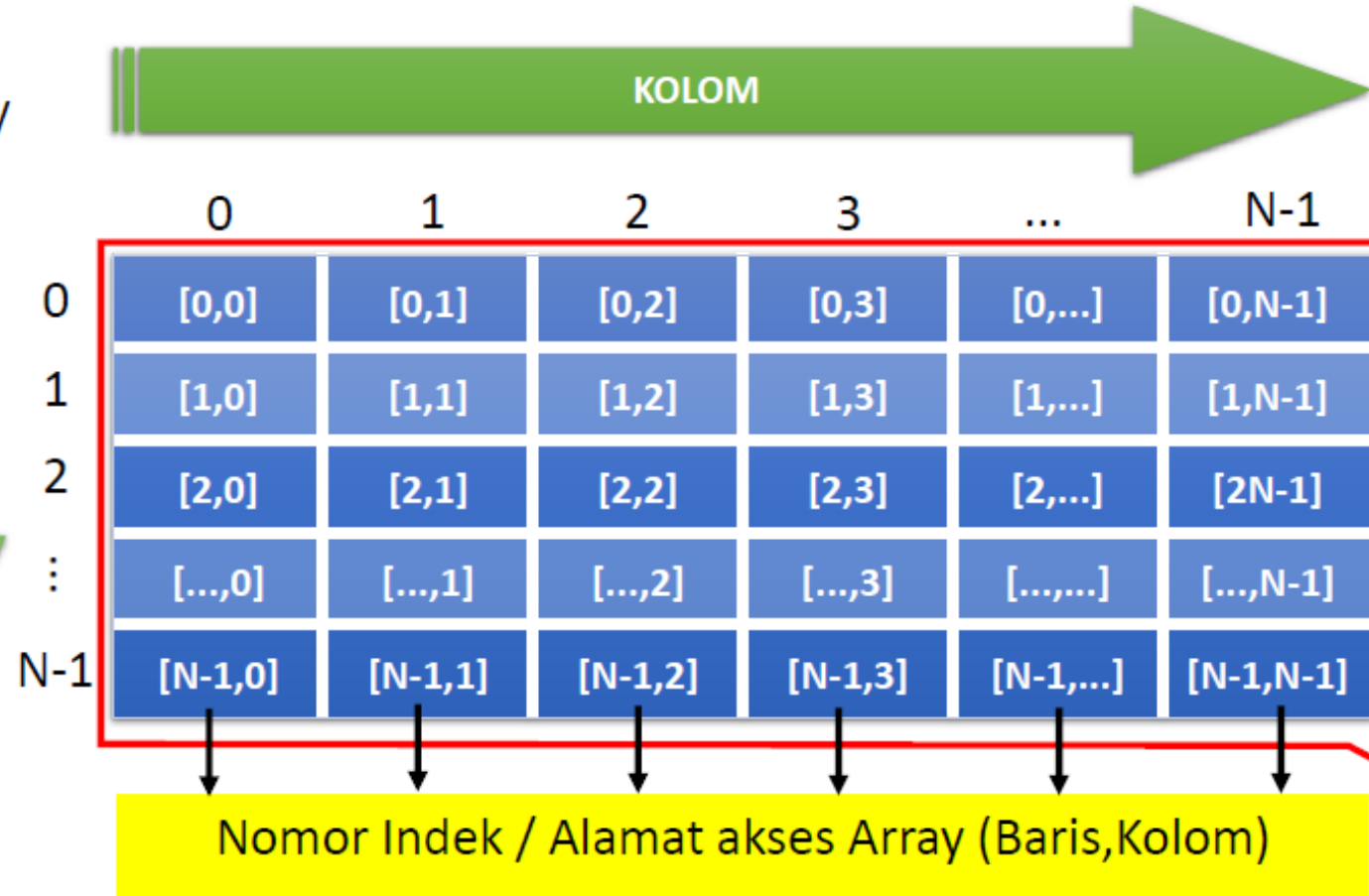
- **Array Multi Dimensi**, array yang mempunyai lebih dari sebuah nomor indeks pada setiap elemen array.
  - ✓ Jika **2 Dimensi** dapat diilustrasikan seperti sebuah **tabel**.
  - ✓ Jika **3 Dimensi** dapat diilustrasikan seperti sebuah **kubus/balok**.
  - ✓ Jika **4 Dimensi** atau **lebih** dapat diilustrasikan seperti penulisan **bab** pada sebuah **buku**. Contoh 3.1.2.3 ... 3.1.2.4 ... 3.1.2.5. dst.



## 2.2. ARRAY DUA DIMENSI

- Anatomi array 2 Dimensi

Tipe data :  
Homogen  
(Integer, Real,  
Karakter,  
Boolean)



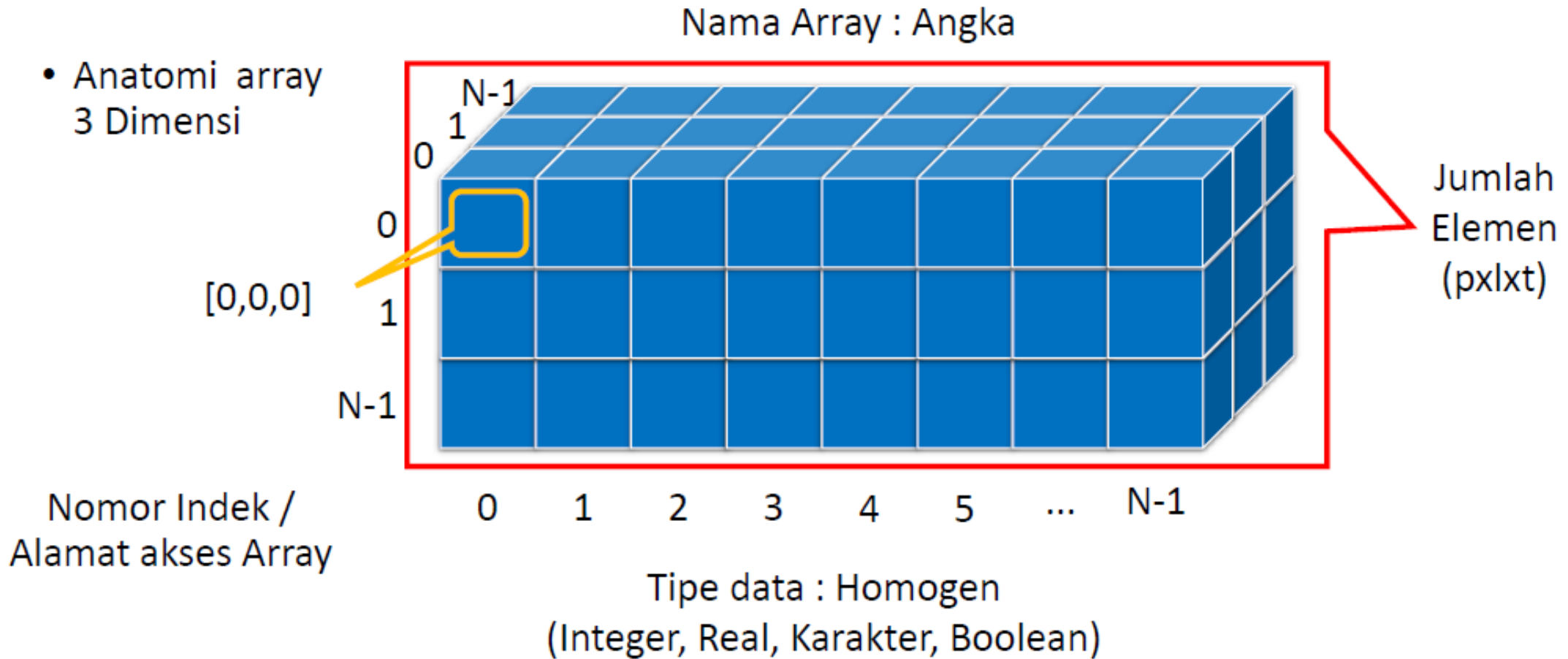
Jumlah  
Elemen





## 2.3. ARRAY TIGA DIMENSI

- Anatomi array 3 Dimensi



## **3. DEKLARASI DAN INISIALISASI ARRAY MULTI DIMENSI**



## 3.1. DEKLARASI DAN INISIALISASI ARRAY 2 DIMENSI (1)

angka		KOLOM	
		0	1
BARIS	0	0	3
	1	1	4
	2	2	5

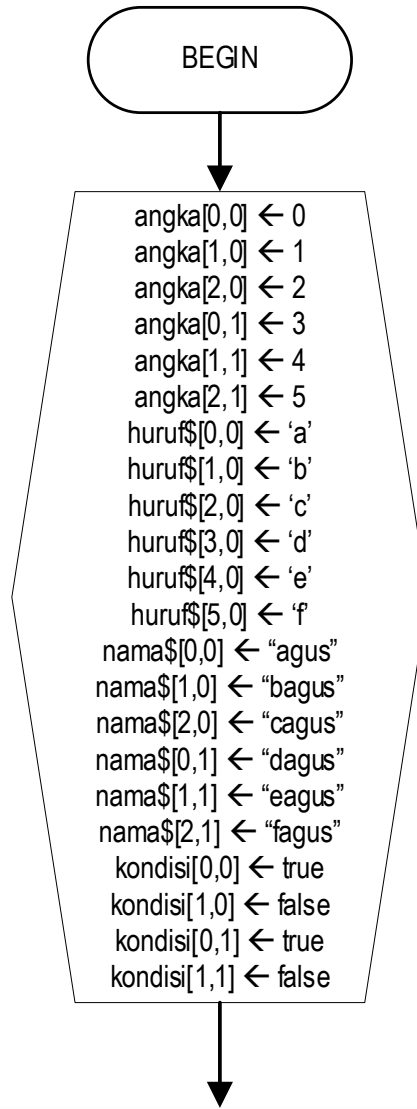
huruf		KOLOM	
		0	1
BARIS	0	a	d
	1	b	e
	2	c	f

nama		KOLOM	
		0	1
BARIS	0	agus	dagus
	1	bagus	eagus
	2	cagus	fagus

kondisi		KOLOM	
		0	1
BARIS	0	true	true
	1	false	false



1. Secara Teori, nomor indeks dimulai dari angka 1
- ✓ **Tips dan trik**, jika dalam deklarasi isi dari variabel array yang memiliki nilai yang sama maka bisa menggunakan looping / perulangan. Hal ini untuk mempersingkat langkah.



## 3.1. DEKLARASI DAN INISIALISASI ARRAY 2 DIMENSI (2)

angka		KOLOM	
		1	2
BARIS	1	0	3
	2	1	4
	3	2	5

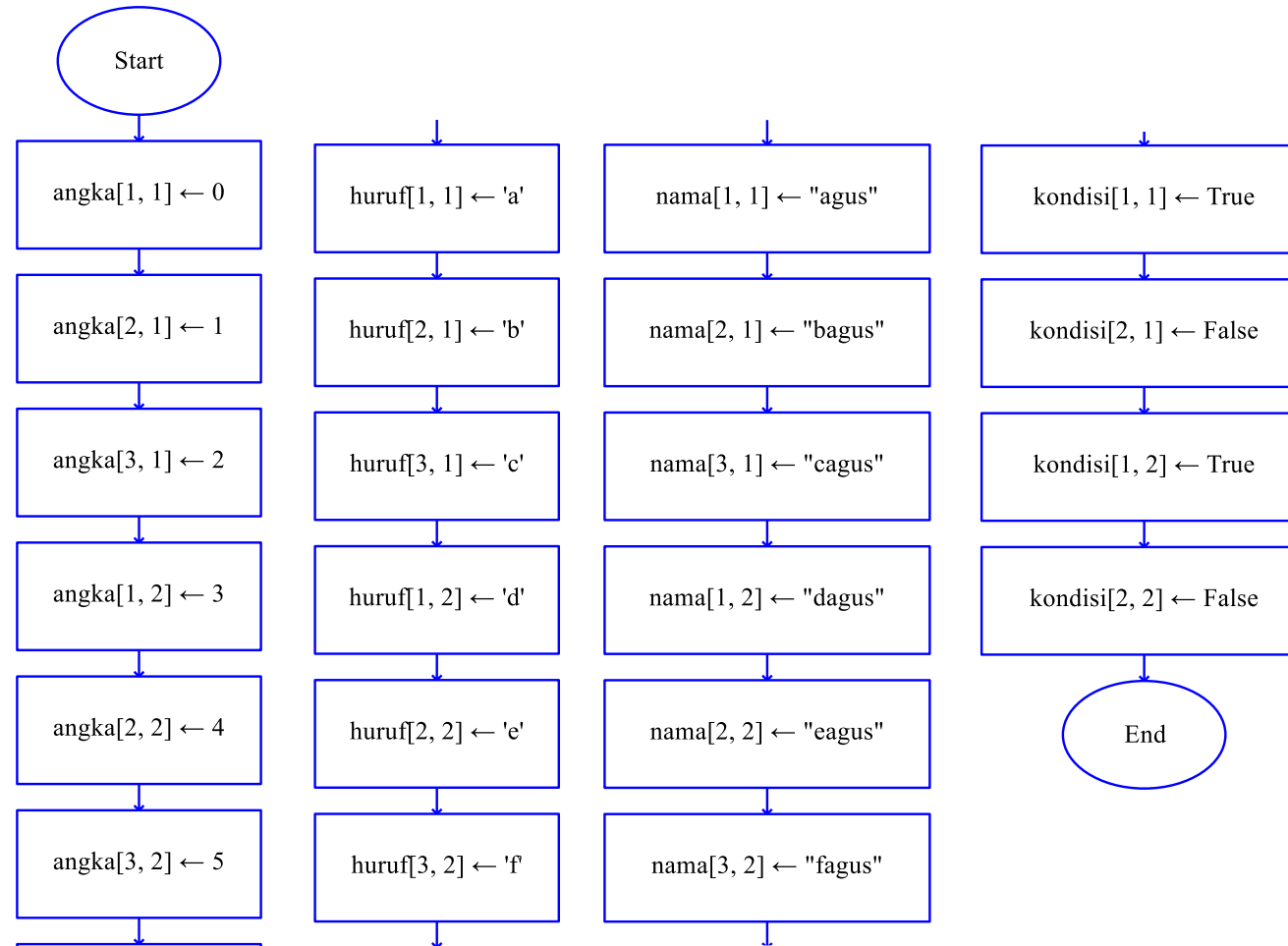
huruf		KOLOM	
		1	2
BARIS	1	a	d
	2	b	e
	3	c	f

nama		KOLOM	
		1	2
BARIS	1	agus	dagus
	2	bagus	eagus
	3	cagus	fagus

kondisi		KOLOM	
		1	2
BARIS	1	true	true
	2	false	false



2. Dalam Raptor, nomor indeks dimulai dari angka 1, simbol sama dengan simbol proses



## 3.2. DEKLARASI DAN INISIALISASI ARRAY 3 DIMENSI (1)

angka				
Z = 0	KOLOM (Y)			
	0	1	2	
BARIS (X)	0	0	3	6
	1	1	4	7
	2	2	5	8

angka				
Z = 1	KOLOM (Y)			
	0	1	2	
BARIS (X)	0	9	12	15
	1	10	13	16
	2	11	14	17

angka				
Z = 2	KOLOM (Y)			
	0	1	2	
BARIS (X)	0	18	21	24
	1	19	22	25
	2	20	23	26

BEGIN

```
angka[0,0,0] ← 0
angka[1,0,0] ← 1
angka[2,0,0] ← 2
angka[0,1,0] ← 3
angka[1,1,0] ← 4
angka[2,1,0] ← 5
angka[0,2,0] ← 6
angka[1,2,0] ← 7
angka[2,2,0] ← 8
```

```
angka[0,0,1] ← 9
angka[1,0,1] ← 10
angka[2,0,1] ← 11
angka[0,1,1] ← 12
angka[1,1,1] ← 13
angka[2,1,1] ← 14
angka[0,2,1] ← 15
angka[1,2,1] ← 16
angka[2,2,1] ← 17
```

```
angka[0,0,2] ← 18
angka[1,0,2] ← 19
angka[2,0,2] ← 20
angka[0,1,2] ← 21
angka[1,1,2] ← 22
angka[2,1,2] ← 23
angka[0,2,2] ← 24
angka[1,2,2] ← 25
angka[2,2,2] ← 26
```

huruf\$				
Z = 0	KOLOM (Y)			
	0	1	2	
BARIS (X)	0	a	d	g
	1	b	e	h
	2	c	f	i

huruf\$				
Z = 1	KOLOM (Y)			
	0	1	2	
BARIS (X)	0	j	m	p
	1	k	n	q
	2	l	o	r

huruf\$				
Z = 2	KOLOM (Y)			
	0	1	2	
BARIS (X)	0	s	v	y
	1	t	w	z
	2	u	x	A

BEGIN

```
huruf$[0,0,0] ← 'a'
huruf$[1,0,0] ← 'b'
huruf$[2,0,0] ← 'c'
huruf$[0,1,0] ← 'd'
huruf$[1,1,0] ← 'e'
huruf$[2,1,0] ← 'f'
huruf$[0,2,0] ← 'g'
huruf$[1,2,0] ← 'h'
huruf$[2,2,0] ← 'i'
```

```
huruf$[0,0,1] ← 'j'
huruf$[1,0,1] ← 'k'
huruf$[2,0,1] ← 'l'
huruf$[0,1,1] ← 'm'
huruf$[1,1,1] ← 'n'
huruf$[2,1,1] ← 'o'
huruf$[0,2,1] ← 'p'
huruf$[1,2,1] ← 'q'
huruf$[2,2,1] ← 'r'
```

```
huruf$[0,0,2] ← 's'
huruf$[1,0,2] ← 't'
huruf$[2,0,2] ← 'u'
huruf$[0,1,2] ← 'v'
huruf$[1,1,2] ← 'w'
huruf$[2,1,2] ← 'x'
huruf$[0,2,2] ← 'y'
huruf$[1,2,2] ← 'z'
huruf$[2,2,2] ← 'A'
```

- ✓ Pada Array 3 dimensi, nomor indeks mengikuti layaknya seperti bangun tiga dimensi sumbu x, y, z

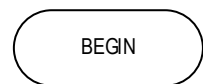


## 3.2. DEKLARASI DAN INISIALISASI ARRAY 3 DIMENSI (2)

nama\$				
Z = 0		KOLOM (Y)		
		0	1	2
BARIS (X)	0	agus	dagus	gagus
	1	bagus	eagus	hagus
	2	cagus	fagus	iagus

nama\$				
Z = 1		KOLOM (Y)		
		0	1	2
BARIS (X)	0	jagus	magus	pagus
	1	kagus	nagus	qagus
	2	lagus	oagus	ragus

nama\$				
Z = 2		KOLOM (Y)		
		0	1	2
BARIS (X)	0	sagus	vagus	yagus
	1	tagus	wagus	zagus
	2	uagus	xagus	AGUS



```

nama$[0,0,0] ← "agus"
nama$[1,0,0] ← "bagus"
nama$[2,0,0] ← "cagus"
nama$[0,1,0] ← "dagus"
nama$[1,1,0] ← "eagus"
nama$[2,1,0] ← "fagus"
nama$[0,2,0] ← "gagus"
nama$[1,2,0] ← "hagus"
nama$[2,2,0] ← "iagus"

```

```

nama$[0,0,1] ← "jagus"
nama$[1,0,1] ← "kagus"
nama$[2,0,1] ← "lagus"
nama$[0,1,1] ← "magus"
nama$[1,1,1] ← "nagus"
nama$[2,1,1] ← "oagus"
nama$[0,2,1] ← "pagus"
nama$[1,2,1] ← "qagus"
nama$[2,2,1] ← "ragus"

```

```

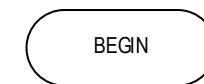
nama$[0,0,2] ← "sagus"
nama$[1,0,2] ← "tagus"
nama$[2,0,2] ← "uagus"
nama$[0,1,2] ← "vagus"
nama$[1,1,2] ← "wagus"
nama$[2,1,2] ← "xagus"
nama$[0,2,2] ← "yagus"
nama$[1,2,2] ← "zagus"
nama$[2,2,2] ← "AGUS"

```

kondisi				
Z = 0		KOLOM (Y)		
		0	1	2
BARIS (X)	0	true	true	true
	1	false	false	false

kondisi				
Z = 1		KOLOM (Y)		
		0	1	2
BARIS (X)	0	true	true	true
	1	false	false	false

kondisi				
Z = 2		KOLOM (Y)		
		0	1	2
BARIS (X)	0	true	true	true
	1	false	false	false



```

kondisi[0,0,0] ← true
kondisi[1,0,0] ← false
kondisi[0,1,0] ← true
kondisi[1,1,0] ← false
kondisi[0,2,0] ← true
kondisi[1,2,0] ← false

```

```

kondisi[0,0,1] ← true
kondisi[1,0,1] ← false
kondisi[0,1,1] ← true
kondisi[1,1,1] ← false
kondisi[0,2,1] ← true
kondisi[1,2,1] ← false

```

```

kondisi[0,0,2] ← true
kondisi[1,0,2] ← false
kondisi[0,1,2] ← true
kondisi[1,1,2] ← false
kondisi[0,2,2] ← true
kondisi[1,2,2] ← false

```

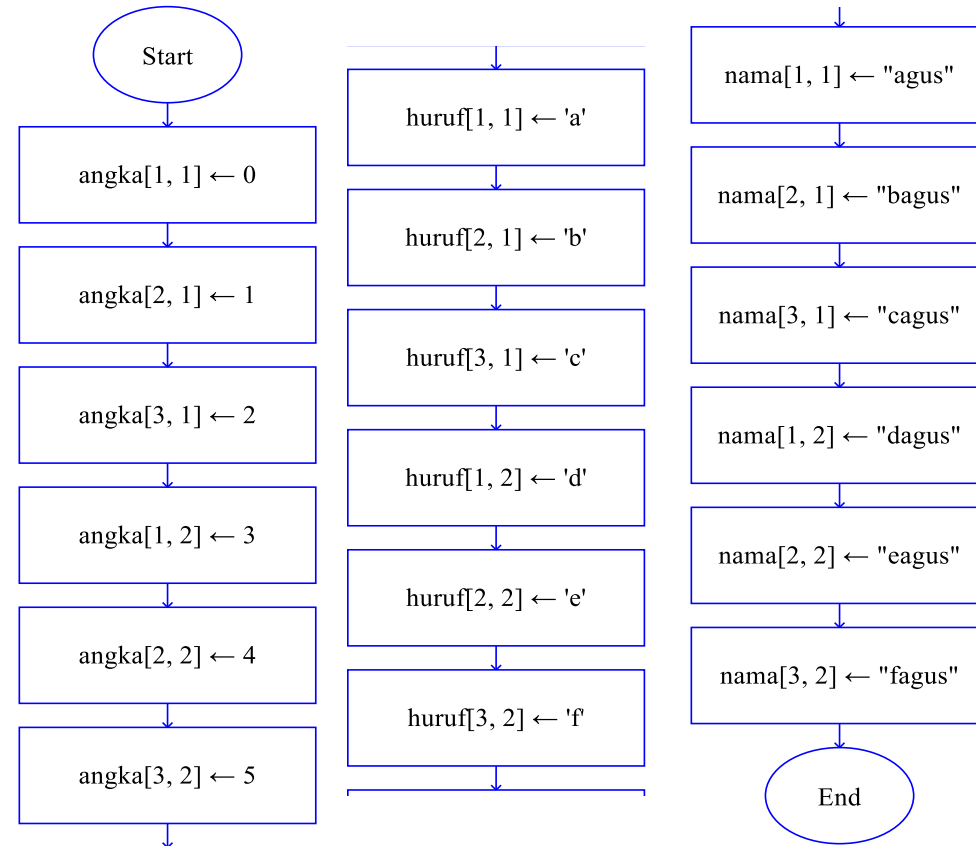


# 4. ASSIGNMENT ARRAY MULTI DIMENSI



## 4.1. ASSIGNMENT ARRAY 2 DIMENSI (1)

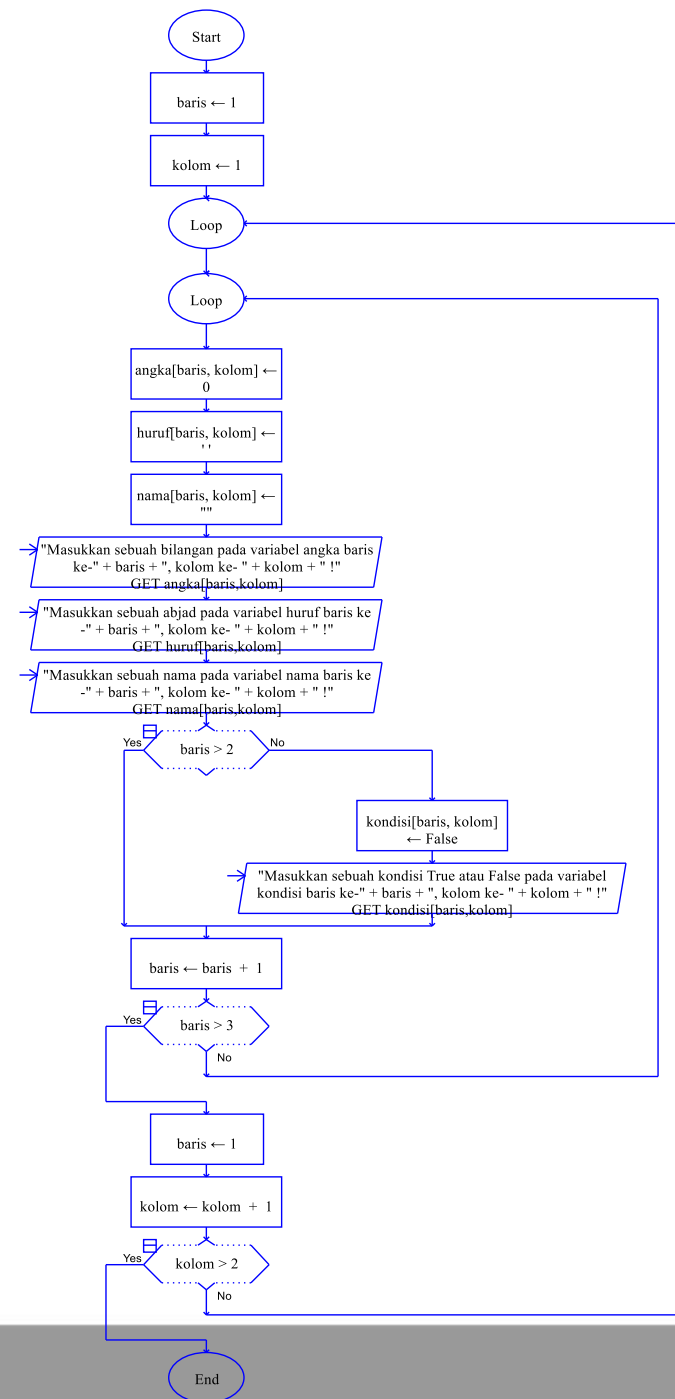
### 1. Dalam tahapan proses





## 4.1. ASSIGNMENT ARRAY 2 DIMENSI (2)

### 2. Dalam tahapan input



## **5. PENGOPERASIAN ARRAY MULTI DIMENSI**



## 5.1. PENGOPERASIAN ARRAY MULTI DIMENSI TIPE DATA NUMERIK

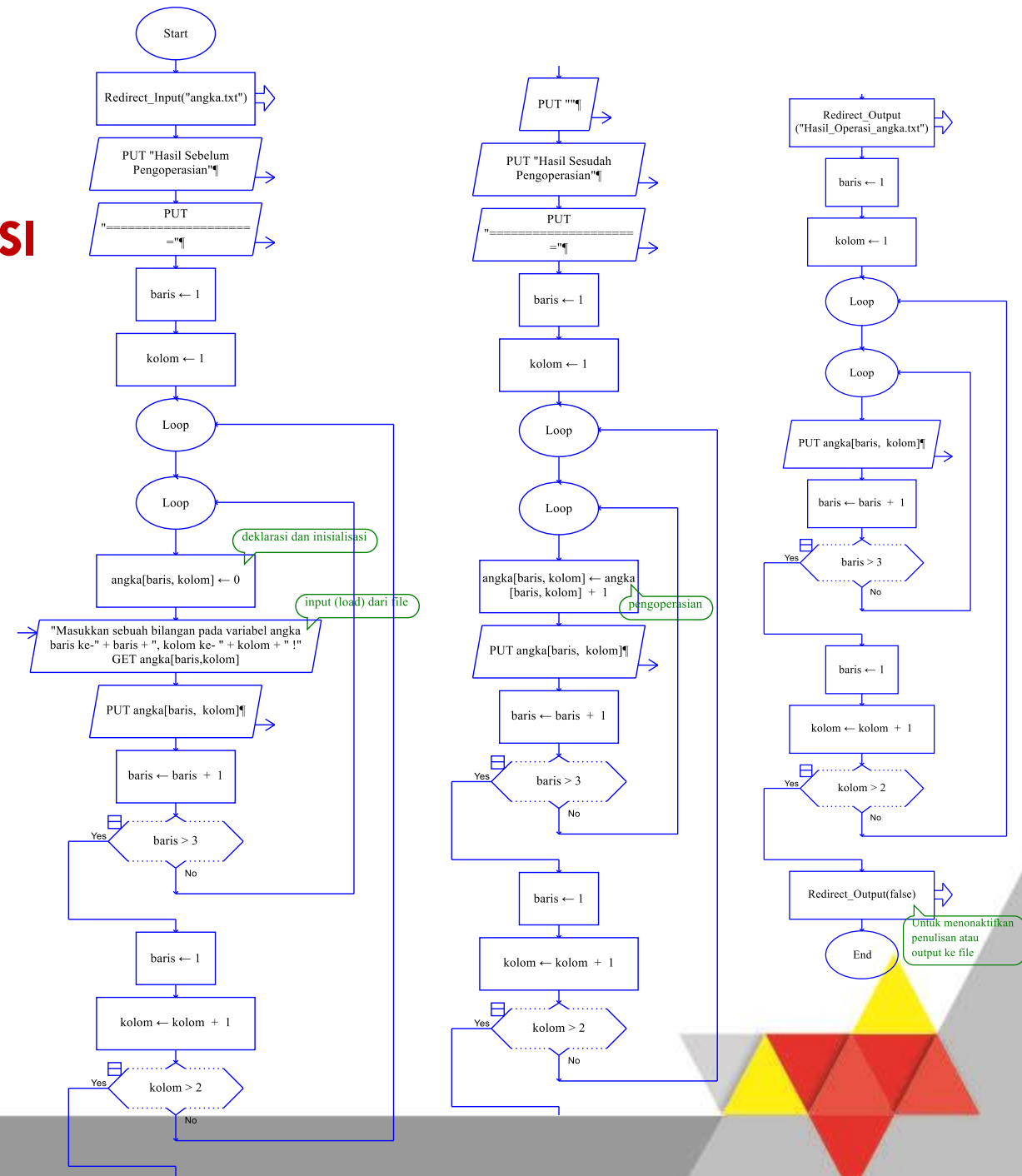
- Operasi penjumlahan isi variabel array 2 dimensi pada setiap elemen + 1
- $\text{angka}[\text{baris}, \text{kolom}] \leftarrow \text{angka}[\text{baris}, \text{kolom}] + 1$

Sebelum

angka		KOLOM	
		1	2
BARIS	1	0	3
	2	1	4
	3	2	5

angka		KOLOM	
		1	2
BARIS	1	1	4
	2	2	5
	3	3	6

Sesudah



## 5.2. PENGOPERASIAN ARRAY MULTI DIMENSI TIPE DATA KARAKTER

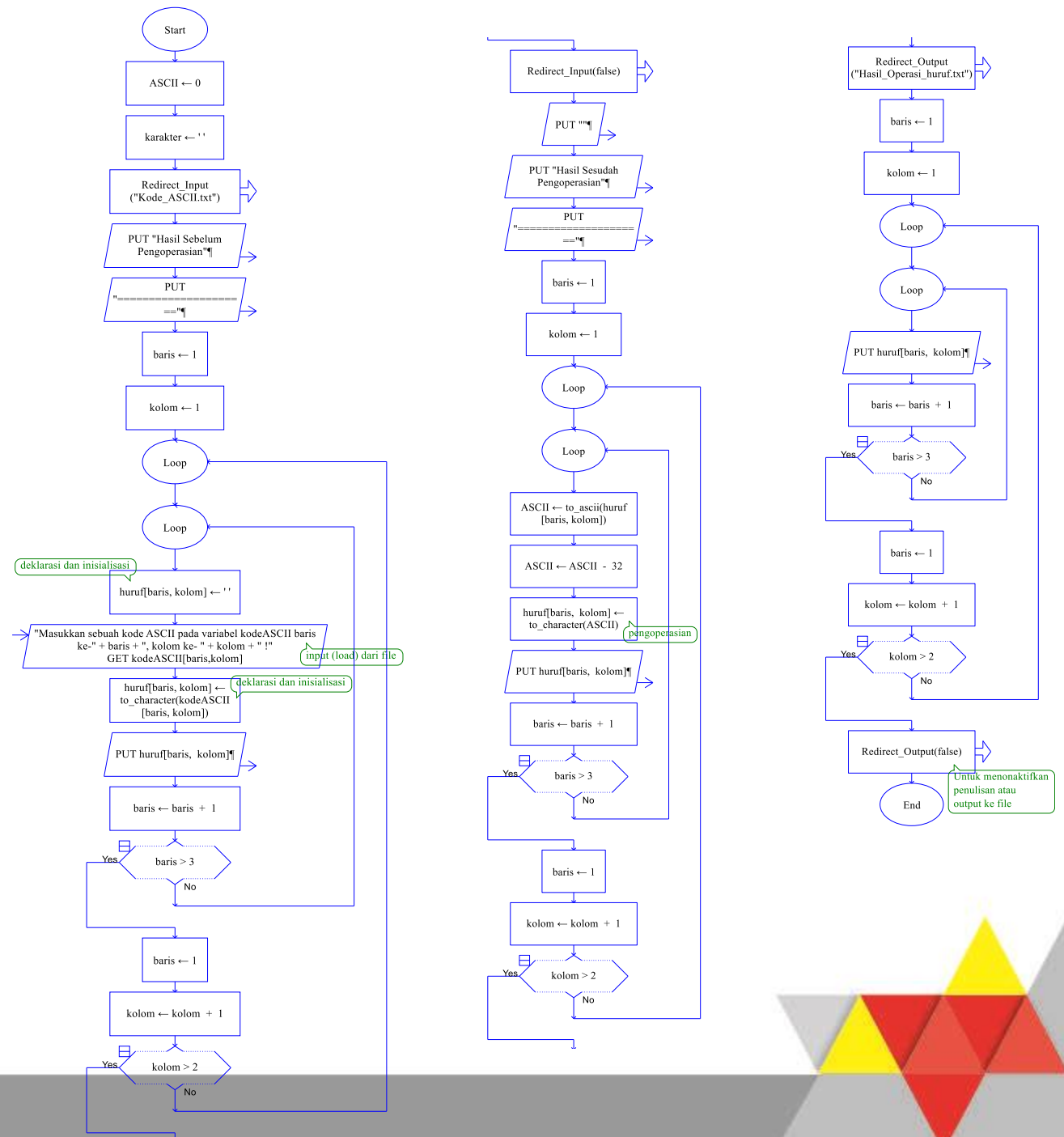
- Operasi penjumlahan isi variabel array 2 dimensi pada setiap elemen Upper Case (menjadikan karakter huruf besar) kode ASCII - 32
- $ASCII \leftarrow to\_ascii(huruf[baris, kolom])$
- $ASCII \leftarrow ASCII - 32$
- $huruf[baris, kolom] \leftarrow to\_character(ASCII)$

Sebelum

		KOLOM	
		1	2
BARIS	1	a	d
	2	b	e
	3	c	f

Huruf\$		KOLOM	
		1	2
BARIS	1	A	D
	2	B	E
	3	C	F

Sesudah

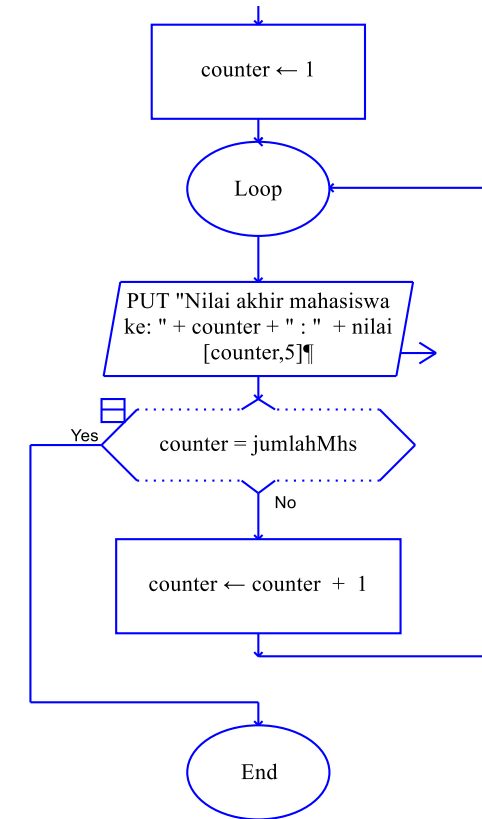
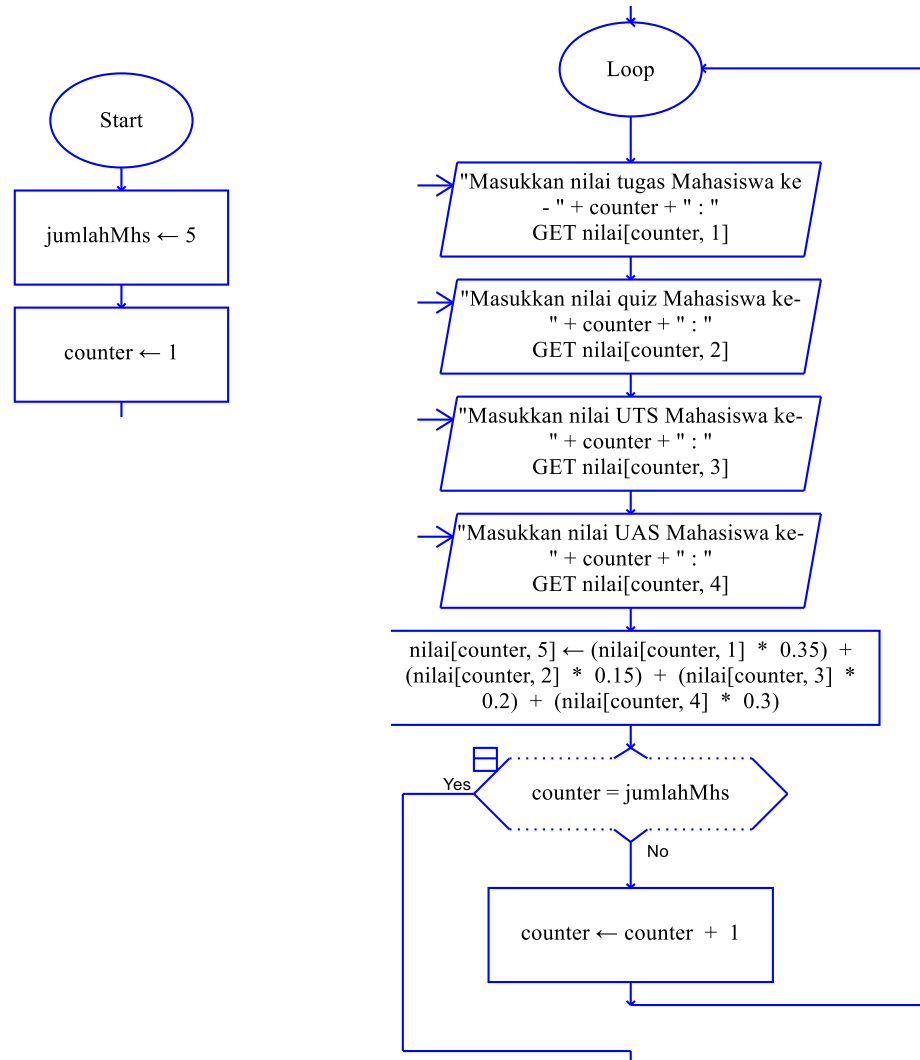


## **6. CONTOH PENERAPAN ARRAY MULTI DIMENSI**



## 6. CONTOH PENERAPAN ARRAY 1 DIMENSI

- Flowchart algoritma menghitung nilai akhir mahasiswa



- Keunggulan array:
  1. Array sangat cocok untuk pengaksesan acak. Sembarang elemen di array dapat di acu secara langsung tanpa melalui elemen-elemen lain.
  2. Jika telah berada di suatu lokasi elemen, maka sangat mudah menelusuri ke elemen-elemen sebelahnya, baik elemen sebelum atau elemen sesudahnya.



- Kelemahan array:
  1. Array mempunyai fleksibilitas rendah, karena array mempunyai batasan harus bertipe homogen. Kita tidak dapat mempunyai array dimana satu elemen adalah karakter, elemen lain bilangan, dan elemen lain adalah tipe-tipe lain
  2. Kebanyakan bahasa pemrograman mengimplementasikan array dengan ukuran statik yang sulit diubah ukurannya diwaktu eksekusi. Jika penambahan dan pengurangan terjadi terus-menerus, maka representasi statis ini bersifat tidak efisien dalam penggunaan memori





1. Ubahlah sebuah variabel dalam tugas / UTS anda menggunakan variabel array 1 dan 2 Dimensi!
2. Tandai tahapan deklarasi dan inisialisasi pada simbol proses dengan pemberian komentar variabel array 1 dan 2 Dimensi pada soal nomor 1
3. Mengikuti soal nomor 1, gunakan:
  - a. Input
    - 1) Dari user
    - 2) Dari load file
  - b. Proses menggunakan operator yang bersesuaian
  - c. Output dalam variabel array 1 dan 2 dimensi tersebut.
    - 1) Ke Console
    - 2) Ke file



- Arrosyidi, A. (2017). In I. B. Surabaya, Buku Ajar Bahasa Pemrograman Java (p. 30). Surabaya: PT Revka Petra Media.
- Institut Bisnis dan Informatika Stikom Surabaya. (2010). Logika dan Algoritma. Surabaya: Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM).



**SELESAI  
TERIMA KASIH**

