



Array

Teknik Informatika

Politeknik Negeri Batam

Pendahuluan

Sebuah variabel hanya menyimpan sebuah nilai

$A \leftarrow 10$

Kebutuhan mengolah sekumpulan data yang bertipe sama

- Hasil ujian 100 mahasiswa
- Tabel harga barang di pasar swalayan
- $a_1, a_2, a_3, a_4, \dots, a_n$

Di pemrograman, kumpulan data bertipe sama perlu disimpan sementara di memori

Array, larik, tabel, vektor, matrik

Array

Struktur data yang menyimpan sekumpulan elemen yang bertipe sama

Setiap elemen diakses langsung melalui indeksnya

Indeks harus mempunyai tipe data yang menyatakan keterurutan

- Integer
- karakter

Array

	A
1	158
2	157
3	162
4	169
5	172
6	155
7	170
8	163

Isi A[1] adalah 158

Isi A[2] adalah 157

...

Isi A[8] adalah 163

Struktur data statik :
jumlah elemen harus
sudah diketahui sebelum
program dieksekusi
+
Jumlah elemen bersifat
tetap

Arti Deklarasi Array

Mendefinisikan banyaknya elemen array

- memesan sejumlah tempat di memori

Mendefinisikan tipe elemen array

- Tipe sederhana + bentukan

A : array [1..100] of integer

Nama: array[1..12] of string

Const Nmaks=100

NilUjian: array [0..Nmaks] of real

$A[4] \leftarrow 10$
Read ($A[4]$)

$i \leftarrow 1$
Nama[i] \leftarrow 'Erna'
Write (Nama[i])

Cara mengacu
elemen array

Program PemrosesanArray

Deklarasi :

const Nmaks=100

type LarikInt : array [1..Nmaks] of integer

A: LarikInt

i: integer

Inisialisasi

$i \leftarrow 1$

While $i \leq Nmaks$ do

pemrosesan terhadap A[i]

$i \leftarrow i+1$

endwhile

terminasi

Inisialisasi

For $i \leftarrow 1$ to Nmaks do

pemrosesan terhadap

A[i]

endfor

terminasi

Mengisi elemen array dengan pembacaan

V1 : jumlah elemen efektif ditentukan di awal

- for $i \leftarrow 1$ to n do
- read(A[i])
- endfor

V2 : jumlah elemen efektif baru diketahui di akhir pembacaan

- $n \leftarrow 0$
- repeat
- $n \leftarrow n + 1$
- read(A[n])
- write('Lagi(y/t)')
- read(jawab)
- Until jawab='t'

Program Rata2TanpaArray

Deklarasi

x1,x2,x3,x4,x5,x6:integer

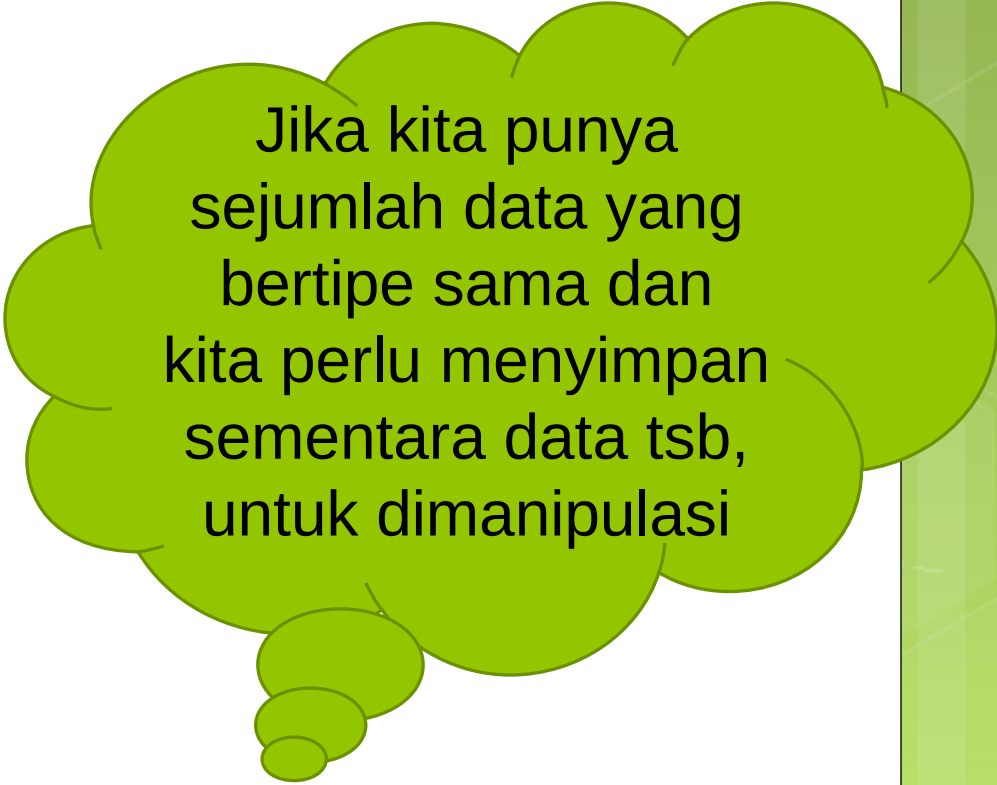
u:real

Algoritma

read(x1,x2,x3,x4,x5,x6)

$u \leftarrow (x1+x2+x3+x4+x5+x6)/6$

write(u)



Jika kita punya sejumlah data yang bertipe sama dan kita perlu menyimpan sementara data tsb, untuk dimanipulasi

Kapan Menggunakan Array?

Program RataRata

Deklarasi

```
const Nmaks=100
```

```
type LarikInt:array[1..Nmaks]  
of real
```

```
A: LarikInt
```

```
i,n: integer
```

```
u,jumlah:real
```

Algoritma

```
//isi array
```

```
read(n) //jumlah data
```

```
for i←1 to n do
```

```
  read(A[i])
```

```
endfor
```

```
//hitung rata-rata
```

```
jumlah←0
```

```
for i←1 to n do
```

```
  jumlah←jumlah + A[i]
```

```
endfor
```

```
u←jumlah/n
```

```
write(u)
```

Mencari Nilai Maksimum Array

Juara kelas

Mahasiswa
paling tinggi

Array terurut
menaik →
elemen
terakhir

Array terurut
menurun
→ elemen
pertama

Array
tersusun
acak?

- Asumsi: nilai maksimum sementara(maks) adalah nilai yang sangat kecil(-9999)
- Array dikunjungi mulai dari elemen pertama
- Jika elemen yang dikunjungi lebih besar dari maks, maks diganti dengan elemen tersebut

Mencari Nilai Maksimum Array-v1

Mencari Nilai Maksimum Array-v1

Asumsi:maks=-9999

A[1]>maks? Ya \rightarrow maks \leftarrow A[1]
(maks=158)

A[2]>maks? Tidak \rightarrow maks tidak berubah

A[3]>maks? ... \rightarrow maks ...

A[4]>maks? ... \rightarrow maks ...

A[5]>maks? ... \rightarrow maks ...

A[6]>maks? ... \rightarrow maks ...

A[7]>maks? ... \rightarrow maks ...

A[8]>maks? ... \rightarrow maks ...

	A
1	158
2	157
3	162
4	169
5	172
6	155
7	170
8	163

Maks=172

Mencari Nilai Maksimum Array-v1

Program CariMaks

Deklarasi

```
const Nmaks=100
```

```
type LarikInt:array[1..Nmaks]  
of integer
```

```
A: LarikInt
```

```
i,n,maks: integer
```

Algoritma

```
//isi array
```

```
read(n)
```

```
for i←1 to n do
```

```
    read(A[i])
```

```
endfor
```

```
//cari nilai maksimum
```

```
Maks ← -9999
```

```
For i←1 to n do
```

```
    if A[i]>maks then
```

```
        maks ← A[i]
```

```
    endif
```

```
endfor
```

```
write(maks)
```

Kelemahan : kasus ada
elemen yang bernilai
kurang dari -9999

Mencari Nilai Maksimum Array-v2

- Maks diinisialisasi dengan elemen pertama array
- Array dikunjungi mulai dari elemen kedua

	A
1	100
2	87
3	162
4	150
5	240
6	202
7	90
8	163

Asumsi:maks=A[1]

A[2]>maks? → maks

A[3]>maks? → maks

A[4]>maks? → maks

A[5]>maks? → maks

A[6]>maks? → maks

A[7]>maks? → maks

A[8]>maks? → maks

Maks =

Mencari Nilai Maksimum Array-v2

Program CariMaksV2

Deklarasi

const Nmaks=100

type LarikInt:array[1..Nmaks]
of integer

A: LarikInt

i,n,maks: integer

Algoritma

//isi array

read(n)

for i←1 to n do

read(A[i])

endfor

//cari nilai maksimum

Maks ← A[1]

For i←2 to n do

if A[i]>maks then

maks ← A[i]

endif

endfor

write(maks)

Mencari Nilai Maksimum Array-v3: Mencari posisi atau indeks elemen array yang bernilai maksimum

Program CariMaksV3

Deklarasi

```
const Nmaks=100
```

```
type LarikInt:array[1..Nmaks]  
of integer
```

```
A: LarikInt
```

```
i,n,IdMaks: integer
```

Algoritma

```
//isi array
```

```
read(n)
```

```
for i←1 to n do
```

```
    read(A[i])
```

```
endfor
```

```
//cari indeks elemen  
    bernilai maksimum
```

```
IdMaks ← . . . .
```

```
For i←2 to n do
```

```
    if A[i]>A[IdMaks] then
```

```
        IdMaks ← . . . .
```

```
    endif
```

```
endfor
```

```
write(IdMaks)
```

Mencari nilai tertentu

Mencari
kata di
kamus

Mencari
nomor
telpon di hp

Mencari
alamat di
perumahan

Pencarian
beruntun

Sampai nilai
yang dicari
ditemukan

Sampai
seluruh elemen
array diperiksa

Algoritma pencarian V1

tanpa menggunakan variabel boolean

Program CariX_V1

Deklarasi

const Nmaks=100

type LarikInt:array[1..Nmaks]
of integer

A: LarikInt

i,n,x,idx: integer

Algoritma

//isi array

read(n)

for i ← 1 to n do

read(A[i])

endfor

Write('Nilai yang dicari :')

Read(x)

i ← 1

while i < n and A[i] ≠ x do

i ← i+1

Endwhile

if A[i]=x then

Idx ← i

write('ditemukan di indeks ke', idx)

else

write('tidak ditemukan')

endif

Algoritma pencarian V2

dengan menggunakan variabel boolean

Program CariX_V2

Deklarasi

const Nmaks=100

type LarikInt:array[1..Nmaks] of
integer

A: LarikInt

i,n,x,idx: integer

ketemu : boolean

Algoritma

read(n)

for i←1 to n do

read(A[i])

endfor

Write('Nilai yang dicari :')

Read(x)

i ← 1

Ketemu ← false

while i ≤ n and not ketemu do

if A[i] = x then

ketemu ← true

else

i ← i + 1

endif

Endwhile

if ketemu then

Idx ← i

write('ditemukan di indeks ke', idx)

else

write('tidak ditemukan')

endif

Kasus Lain

Menyalin array

- Ada array A yang berisi n elemen
- Elemen array A kita salin ke array B

Menguji kesamaan 2 buah array

- Ada dua array A dan B, yang jumlah elemennya sama
- Dua array dikatakan sama bila elemen2 yang bersepadanan urutannya sama.
- Jika A dan B sama, algoritma menghasilkan nilai true

Kasus Lain

A	
1	100
2	87
3	162
4	150
5	240
6	202
7	90
8	163

B	
1	100
2	87
3	162
4	150
5	240
6	202
7	90
8	163

C	
1	100
2	87
3	162
4	150
5	245
6	202
7	90
8	163

Array Bertipe Terstruktur

	NIM	Nama	IPK
1	298001	Heru	3.5
2	298002	Ika	2.87
3	298003	Maya	3.45
4	298004	Santi	3.8
5			
6			
.			
.			
100	298100	Soni	2.77

Isi record :
NIM
Nama
IPK

Array Bertipe Terstruktur

Deklarasi :

```
const Nmaks=100
```

```
type Mahasiswa : record <NIM : string,  
                        nama : string,  
                        IPK : real>
```

```
type TabMhs:array[1..Nmaks] of Mahasiswa  
Mhs: TabMhs
```


Array Bertipe Terstruktur

Tidak boleh

- `Mhs[2]`
- `Write(Mhs[i])`

Boleh

- `Write(Mhs[2].NIM)`
- `Write(Mhs[2].nama)`
- `Write(Mhs[i].IPK)`

String sebagai array karakter

- Definisi : array karakter dengan panjang dinamis
- Karena string adalah array, maka elemen-elemen string diakses melalui indeks
- S: string
- $S \leftarrow$ 'Ini Budi'
- Panjang : 8
- Elemen-elemennya :
 - S[1]='I'
 - S[2]='n'
 - S[3]='i'
 - S[4]=' '
 - S[5]='B'
 - S[6]='u'
 - S[7]='d'
 - S[8]='i'

Tugas

Buat alat peraga/video/animasi yang menggambarkan algoritma di bawah ini. Jika algoritma blm tersedia, tulis algoritmanya dulu.
Kumpulkan laporan dan hasil minggu depan(learning da didemokan)

Kelompok	Deskripsi
Kel1	menghitung rata-rata pada array
Kel2	Mencari Nilai Maksimum Array-v1
Kel3	Mencari Nilai Maksimum Array-v2
Kel4	Mencari Nilai Maksimum Array-v3
kel5	Mencari nilai tertentu(tanpa menggunakan variabel boolean)
Kel6	Mencari nilai tertentu(dengan menggunakan variabel boolean)
Kel7	Menyalin array
Kel8	Menguji kesamaan 2 buah array
Kel9	Array Bertipe Terstruktur, untuk mencari status lulus.Jika IPK ≥ 2 , mahasiswa dikatakan lulus
Kel10	Membalik Isi Array

Referensi

- Rinaldi Munir, *Algoritma dan Pemrograman dalam Bahasa Pascal dan C (Edisi Ke-3)*, penerbit Informatika, 2005