



Array

Tim Olimpiade Komputer Indonesia

Pendahuluan

Melalui dokumen ini, kalian akan:

- Memahami konsep **array**.
- Mengimplementasikan **array** pada bahasa Pascal.
- Menggunakan **array** untuk penyelesaian beberapa contoh masalah.



Bagian 1

Konsep Array



Motivasi

- Pak Dengklek memiliki sebuah tumpukan berisi N kartu, yang dipenuhi $1 \leq N \leq 100$.
- Setiap kartu bertuliskan suatu bilangan bulat.
- Sekarang Pak Dengklek ingin tahu urutan angka-angka pada kartu tersebut bila tumpukan kartu itu dibalik.
- Contoh: jika diberikan 5 kartu dengan angka-angka dari atasnya $[1, 5, 3, 20, 4]$, maka setelah dibalik urutannya menjadi: $[4, 20, 3, 5, 1]$.
- Bantulah Pak Dengklek menentukan urutan angka-angka tersebut setelah tumpukan kartu dibalik!



Solusi?

- Sederhana, idenya adalah dengan menampung seluruh bilangan terlebih dahulu, baru dicetak dalam urutan terbalik.
- Misalnya jika N selalu 3, kita bisa membuat 3 variabel (misalnya a , b , c), lalu:

```
readln(a);  
readln(b);  
readln(c);
```

```
writeln(c);  
writeln(b);  
writeln(a);
```

- Sayangnya nilai N tidak tetap! Dibutuhkan suatu mekanisme lain untuk menggunakan dan mengakses variabel!



Pengertian Array

Array

Variabel dengan satu nama, tetapi mengandung banyak nilai.
Akses nilai-nilainya dilakukan dengan indeks.

Perhatikan contoh berikut!

indeks	1	2	3	4	5	6	7	8	9	10
A	3	10	11	23	35	12	31	53	0	19

- $A[1] = 3$
- $A[2] = 10$
- $A[5] = 35$



Penjelasan

- Pada contoh sebelumnya, kita memiliki sebuah variabel bernama **A**.
- **A** memiliki 10 nilai, yang masing-masing dapat diakses dengan indeks.
- Untuk mengakses nilai **A** yang ke- x , digunakan **A[x]**.
- Lebih jauh lagi, sebenarnya **A[x]** bisa dianggap sebagai sebuah variabel yang berdiri sendiri.
- Konsep inilah yang disebut sebagai **array**!



Bagian 2

Implementasi Array Pada Pascal



Deklarasi

- Karena **array** merupakan variabel, diperlukan deklarasi seperti variabel lainnya.
- Format deklarasi **array** adalah:

<nama>: **array**[<nilai awal>..**nilai akhir**] **of** <tipe>;

- Dengan:
 - <nama> adalah nama dari **array** (aturan penamaan sama seperti variabel biasanya)
 - <nilai awal> dan <nilai akhir> adalah rentang indeks **array** yang terdefinisi (boleh saja negatif).
 - <tipe> adalah tipe data dari **array**.
- Tentu saja, tipe data di sini bisa berupa **longint**, **double**, **string**, **boolean** atau suatu **record**.



Contoh Deklarasi

Berikut ini adalah contoh deklarasi **array** pada Pascal:

var

tabel: **array**[0..100] **of** **boolean**;

frekuensi: **array**[-1000..1000] **of** **longint**;

- Untuk contoh **array** **tabel**, hanya tabel[0], tabel[1], tabel[2], ..., tabel[100] yang terdefinisi.
- Mengakses nilai tabel[-1], tabel[-2], atau tabel[500] akan menyebabkan **runtime error**. Sementara mengakses nilai frekuensi[-1] atau frekuensi[-100] sama sekali tidak bermasalah.
- Untuk itu, tentukan rentang indeks yang akan kalian gunakan saat deklarasi dengan tepat (sesuai kebutuhan).



Array dan Variabel

- Karena suatu elemen dari **array** juga bisa dianggap variabel, tentu saja kita bisa melakukan perintah **readln** padanya.
- Sebagai contoh, jika kita memiliki **array** bernama **tabel** yang terdefinisi dari 1 sampai dengan **100**, kita bisa melakukan:

```
readln(tabel[2]);
```



Array dan Variabel (lanj.)

- Jika diberikan **5** bilangan, dan kita perlu menyimpan masing-masing bilangan di tabel, kita bisa melakukan:

```
readln(tabel[1]);  
readln(tabel[2]);  
readln(tabel[3]);  
readln(tabel[4]);  
readln(tabel[5]);
```

- Tentu saja hal ini sangat tidak efisien!
- Untungnya, kita sudah mempelajari sebuah teknik yang sangat penting, yaitu **perulangan**.



Array dan Variabel (lanj.)

- Proses membaca 5 bilangan pada 5 baris kini bisa dilakukan dengan cara:

```
for i := 1 to 5 do begin  
    readln(tabel[i]);  
end;
```

- Untuk kasus umum, yaitu ketika diberikan N bilangan, cukup ganti angka 5 dengan variabel N .

```
for i := 1 to N do begin  
    readln(tabel[i]);  
end;
```



Array dan Variabel (lanj.)

- Demikian pula untuk pencetakan secara terbalik, kita bisa menggunakan perulangan sebagai berikut:

```
for i := N downto 1 do begin  
    writeln(tabel[i]);  
end;
```

- Sekarang masalah Pak Dengklek terpecahkan!



Contoh Solusi: balik.pas

Berikut contoh solusi lengkap untuk permasalahan motivasi:

```
var
  N, i: longint;
  tabel: array[1..100] of longint;

begin
  readln(N);

  for i := 1 to N do begin
    readln(tabel[i]);
  end;

  for i := N downto 1 do begin
    writeln(tabel[i]);
  end;
end.
```



Array dan Memori

- Setiap elemen pada **array** membutuhkan memori, bergantung pada tipe data yang digunakan.
- Total memori yang dibutuhkan untuk sebuah **array** sama dengan banyaknya elemennya dikali ukuran memori satu elemennya.
- Sebagai contoh, **array** dengan 100 elemen dan memiliki tipe **longint** membutuhkan memori sebesar $100 \times 4 \text{ byte} = 400 \text{ byte}$,



Rentang Array

- Pada **balik.pas**, dideklarasikan **array** sebesar 100 elemen (dari 1 sampai dengan 100), padahal bisa jadi hanya digunakan sebagian saja.
- Cara ini memang "boros" memori, tetapi ingat bahwa kita harus mendeklarasikan **array** tersebut di awal, yang mana pada saat itu tidak diketahui berapa nilai N .
- Dengan demikian, cara yang paling mudah adalah mendeklarasikannya sebesar nilai N maksimal yang mungkin.



Contoh Soal: Ujian Harian

Deskripsi:

- Pak Dengklek menyelenggarakan ujian harian setelah selesai mengajarkan N ekor bebeknya mengenai konsep **array**.
- Setiap bebek ke- i mendapatkan nilai sebesar h_i , yang merupakan bilangan bulat.
- Untuk menentukan lulus atau tidaknya seekor bebek, nilai bebek tersebut harus tidak kurang dari nilai rata-rata dari seluruh bebek.
- Tentukan banyaknya bebek yang lulus ujian!

Batasan:

- $1 \leq N \leq 100$
- $1 \leq h_i \leq 100$, untuk $1 \leq i \leq N$



Contoh Soal: Ujian Harian (lanj.)

Format masukan:

- Baris pertama berisi sebuah bilangan bulat N .
- N baris berikutnya berisi nilai ujian bebek. Baris ke- i ini merupakan h_i .

Format keluaran:

- Sebuah baris yang menyatakan banyaknya bebek yang lulus ujian.



Contoh Soal: Ujian Harian (lanj.)

Contoh masukan:

3

5

6

7

Contoh keluaran:

2

Penjelasan

Nilai rata-rata dari seluruh bebek adalah 6, dan terdapat 2 ekor bebek yang nilainya tidak kurang dari 6.



Petunjuk

- Salah satu solusinya adalah melalui dua tahap:
 1. Hitung rata-ratanya.
 2. Hitung banyaknya bebek yang nilainya tidak kurang dari rata-rata.
- Sebisa mungkin, hindari penggunaan **floating-point**!
 - Ingat bahwa tipe data **floating-point** kurang bisa menyatakan bilangan secara akurat; nilai $1/3 \times 3$ bisa jadi 0.9999999999999999 atau 1.00000000000001.
 - Pengoperasian tipe data **integer** oleh komputer jauh lebih cepat daripada pengoperasian tipe data **floating-point**!



Contoh Solusi: lulus.pas

```
var
  N, i, total, lulus: longint;
  h: array[1..100] of longint;
begin
  readln(N);
  for i := 1 to N do begin
    readln(h[i]);
  end;

  total := 0;
  for i := 1 to N do begin
    total := total + h[i];
  end;
```



Contoh Solusi: lulus.pas (lanj.)

```
lulus := 0;
for i := 1 to N do begin
    (* trik menghindari pembagian *)
    if (h[i]*N >= total) then begin
        lulus := lulus + 1;
    end;
end;

writeln(lulus);
end.
```



Bagian 3

Penggunaan Array Lanjutan



Array Dua Dimensi

- Struktur **array** bisa juga membentuk sebuah tabel dua dimensi.
- Perhatikan contoh deklarasi berikut:

```
matriks: array[1..2, 1..5] of longint;
```

- Kini kita mendapatkan variabel bernama *matriks*[*a*][*b*], yang terdefinisi untuk $1 \leq a \leq 2$ dan $1 \leq b \leq 5$.



Array Dua Dimensi (lanj.)

- Akses suatu elemen dapat dilakukan dengan dua cara, yaitu `matriks[a][b]` atau `matriks[a,b]`.
- Tabel berikut menunjukkan struktur dari **array** matriks:

	1	2	3	4	5
1					
2					

- Aturan perhitungan memori tetap sama; banyaknya elemen dikali memori per elemennya.
Pada kasus ini: $2 \times 5 \times 4 \text{ byte} = 40 \text{ byte}$.



Contoh Soal: Cokelat Bebek

Deskripsi:

- Pak Ganesh datang bertamu ke peternakan bebek Pak Dengklek.
- Pada peternakan bebek Pak Dengklek, terdapat kandang bebek yang tersusun atas petak-petak N baris dan N kolom.
- Pak Dengklek memberi $d_{i,j}$ gram cokelat* ke kandang di baris ke- i dan kolom ke- j .
- Pak Ganesh memberi $g_{i,j}$ gram cokelat* ke kandang di baris ke- i dan kolom ke- j .
- Tentukan berapa gram cokelat yang diperoleh setiap bebek di kandangnya!

Batasan:

- $1 \leq N \leq 100$
- $0 \leq d_{i,j}, h_{i,j} \leq 10$, untuk $1 \leq i, j \leq N$

*Catatan: bebek-bebek suka cokelat!



Contoh Soal: Cokelat Bebek (lanj.)

- Sebagai contoh, misalkan $N = 3$.
- Kemudian berikut adalah cokelat yang diberikan Pak Dengklek (D) dan Pak Ganesh (G):

$$D = \begin{bmatrix} 1 & 3 & 0 \\ 6 & 2 & 4 \\ 2 & 1 & 5 \end{bmatrix}$$

$$G = \begin{bmatrix} 2 & 1 & 7 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

- Maka total cokelat yang didapatkan setiap kandang adalah:

$$\begin{bmatrix} 3 & 4 & 7 \\ 6 & 2 & 5 \\ 3 & 2 & 7 \end{bmatrix}$$



Contoh Soal: Cokelat Bebek (lanj.)

Format masukan:

- Baris pertama berisi sebuah bilangan bulat N .
- N baris berikutnya berisi N bilangan. Bilangan di baris ke- i dan kolom ke- j ini adalah $d_{i,j}$.
- N baris sisanya berisi N bilangan. Bilangan di baris ke- i dan kolom ke- j ini adalah $g_{i,j}$.

Format keluaran:

- N baris yang berisi N bilangan. Bilangan di baris ke- i dan kolom ke- j ini adalah total makanan yang ada di kandang baris ke- i dan kolom ke- j .



Contoh Soal: Cokelat Bebek (lanj.)

Contoh masukan:

3

1 3 0

6 2 4

2 1 5

2 1 7

0 0 1

1 1 2

Contoh keluaran:

3 4 7

6 2 5

3 2 7



Petunjuk

- Salah satu cara yang mudah adalah membuat tiga **array** dua dimensi, masing-masing untuk menampung makanan yang diberikan Pak Dengklek (D), Pak Ganesh (G), dan hasil akhirnya ($hasil$).
- Tentu saja hubungannya adalah $hasil[i][j] = D[i][j] + G[i][j]$, untuk $1 \leq i, j \leq N$.



Solusi: cokelat.pas

Pertama, mari kita deklarasikan variabel yang akan digunakan:

var

 N: longint;

 D, G, hasil: array[1..100, 1..100] of longint;

 i, j: longint;



Solusi: cokelat.pas (lanj.)

Kemudian baca masukan sesuai dengan format yang diberikan:

```
begin
  readln(N);
  for i := 1 to N do begin
    for j := 1 to N do begin
      read(D[i][j]);
    end;
    readln;
  end;

  for i := 1 to N do begin
    for j := 1 to N do begin
      read(G[i][j]);
    end;
    readln;
  end;
```



Solusi: coklat.pas (lanj.)

Lakukan penjumlahan, lalu cetak hasilnya:

```
for i := 1 to N do begin
  for j := 1 to N do begin
    hasil[i][j] := D[i][j] + G[i][j];
  end;
end;

for i := 1 to N do begin
  for j := 1 to N do begin
    write(hasil[i][j]);
    if (j < N) then begin
      write(' ');
    end;
  end;
  writeln;
end;
end.
```



Solusi: cokelat_2.pas

Nilai **array** *D* dan *G* sebenarnya tidak perlu disimpan, kita bisa menghemat memori dengan langsung menjumlahkannya.

```
var
    N: longint;
    temp: longint;
    hasil: array[1..100, 1..100] of longint;
    i, j: longint;

begin
    readln(N);

    for i := 1 to N do begin
        for j := 1 to N do begin
            read(temp);
            hasil[i][j] := temp;
        end;
        readln;
    end;
```



Solusi: cokelat_2.pas (lanj.)

```
for i := 1 to N do begin
  for j := 1 to N do begin
    read(temp);
    hasil[i][j] := hasil[i][j] + temp;
  end;
  readln;
end;
```

```
for i := 1 to N do begin
  for j := 1 to N do begin
    write(hasil[i][j]);
    if (j < N) then
      write(' ');
    end;
    writeln;
  end;
end.
```



Array Multidimensi

- Tidak hanya sampai dua dimensi, dimensi tiga, empat, atau lebih pun bisa.
- Sebagai contoh:

```
data: array[1..2, 1..50, 1..50] of longint;
```

- Kita akan mendapatkan variabel $data[i][j][k]$ yang terdefinisi untuk $1 \leq i \leq 2$, dan $1 \leq j, k \leq 50$.
- Akses elemen juga bisa dilakukan dengan $data[i, j, k]$.



Selanjutnya...

- Mempelajari tentang **fungsi** dan **prosedur**.

