



Array

Tim Olimpiade Komputer Indonesia

Pendahuluan

Melalui dokumen ini, kalian akan:

- Memahami konsep array.
- Mengimplementasikan array pada bahasa C++.
- 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:

```
scanf("%d", &a);  
scanf("%d", &b);  
scanf("%d", &c);
```

```
printf("%d\n", c);  
printf("%d\n", b);  
printf("%d\n", 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 C++



Deklarasi

- Karena array merupakan variabel, diperlukan deklarasi seperti variabel lainnya.
- Format deklarasi array adalah:
<tipe> <nama>[<ukuran>];
- Dengan:
 - <nama> adalah nama dari array (aturan penamaan sama seperti variabel biasanya)
 - <ukuran> adalah ukuran dari array, yang terdefinisi dari 0 sampai dengan ukuran-1.
 - <tipe> adalah tipe data dari array.
- Tentu saja, tipe data di sini bisa berupa int, double, string, bool atau suatu struct.



Contoh Deklarasi

Berikut ini adalah contoh deklarasi array pada C++:

```
bool tabel[101];  
int frekuensi[1000];
```

- Untuk contoh array tabel, hanya tabel[0], tabel[1], tabel[2], ..., tabel[100] yang terdefinisi.
- Untuk contoh array frekuensi, hanya frekuensi[0], frekuensi[1], frekuensi[2], ..., frekuensi[999] yang terdefinisi.
- Mengakses nilai tabel[-1], tabel[-2], atau tabel[500] dapat menyebabkan **runtime error**.
- 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 scanf padanya.
- Sebagai contoh, jika kita memiliki array int bernama tabel yang terdefinisi dari 1 sampai dengan 100, kita bisa melakukan:

```
scanf("%d", &tabel[2]);
```



Array dan Variabel (lanj.)

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

```
scanf ("%d", &tabel[0]);  
scanf ("%d", &tabel[1]);  
scanf ("%d", &tabel[2]);  
scanf ("%d", &tabel[3]);  
scanf ("%d", &tabel[4]);
```

- 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 (int i = 0; i < 5; i++) {  
    scanf("%d", &tabel[i]);  
}
```

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

```
for (int i = 0; i < N; i++) {  
    scanf("%d", &tabel[i]);  
}
```



Array dan Variabel (lanj.)

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

```
for (int i = N-1; i >= 0; i--) {  
    printf("%d\n", tabel[i]);  
}
```

- Sekarang masalah Pak Dengklek terpecahkan!



Contoh Solusi: balik.cpp

Berikut contoh solusi lengkap untuk permasalahan motivasi:

```
#include <cstdio>

int main() {
    int N;
    int tabel[100];
    scanf("%d", &N);

    for (int i = 0; i < N; i++) {
        scanf("%d", &tabel[i]);
    }

    for (int i = N-1; i >= 0; i--) {
        printf("%d\n", tabel[i]);
    }
}
```



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 int membutuhkan memori sebesar $100 \times 4 \text{ byte} = 400 \text{ byte}$,



Rentang Array

- Pada program membalik array, dideklarasikan array sebesar 100 elemen (dari 0 sampai dengan 99), padahal bisa jadi hanya digunakan sebagian saja.
- Cara ini memang "boros" memori, tetapi merupakan cara yang paling mudah adalah mendeklarasikannya sebesar nilai N maksimal yang mungkin.
- Bisa juga kita deklarasikan sesudah N diketahui sebagai berikut:

```
...  
int N;  
  
scanf("%d", &N);  
  
int tabel[N];  
...
```



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.
- Tentukan banyaknya bebek yang memiliki nilai tidak kurang dari rata-rata seluruh bebek!

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 bilangan bulat oleh komputer jauh lebih cepat daripada pengoperasian tipe data floating-point!



Contoh Solusi: lulus.cpp

```
#include <stdio>

int main() {
    int N;
    scanf("%d", &N);

    int nilai[N];
    for (int i = 0; i < N; i++) {
        scanf("%d", &nilai[i]);
    }

    int total = 0;
    for (int i = 0; i < N; i++) {
        total += nilai[i];
    }
}
```



Contoh Solusi: lulus.cpp (lanj.)

```
int lulus = 0;
for (int i = 0; i < N; i++) {
    // Trik menghindari pembagian
    if (nilai[i]*N >= total) {
        lulus++;
    }
}

printf("%d\n", lulus);
}
```



Bagian 3

Penggunaan Array Lanjutan



Array Dua Dimensi

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

```
int matriks[2][5];
```

- Kini kita mendapatkan variabel bernama *matriks[a][b]*, yang terdefinisi untuk $0 \leq a \leq 1$ dan $0 \leq b \leq 4$.



Array Dua Dimensi (lanj.)

- Akses suatu elemen dapat dilakukan dengan `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):

$$D = \begin{bmatrix} 1 & 3 & 0 \\ 6 & 2 & 4 \\ 2 & 1 & 5 \end{bmatrix} \qquad G = \begin{bmatrix} 2 & 1 & 7 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

- 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: coklat.cpp

Pertama, mari kita deklarasikan variabel dan baca masukan:

```
#include <stdio>
```

```
int main() {
    int N;
    scanf("%d", &N);

    int D[N][N], G[N][N], hasil[N][N];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            scanf("%d", &D[i][j]);
        }
    }

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            scanf("%d", &G[i][j]);
        }
    }
}
```



Solusi: cokelat.cpp (lanj.)

Lakukan penjumlahan, lalu cetak hasilnya:

```
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        hasil[i][j] = D[i][j] + G[i][j];
    }
}

for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        printf("%d", hasil[i][j]);
        if (j+1 < N) {
            printf(" ");
        }
    }
    printf("\n");
}
}
```



Solusi: cokelat_2.cpp

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

```
#include <stdio>

int main() {
    int N;
    scanf("%d", &N);

    int hasil[N][N];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            int temp;
            scanf("%d", &temp);
            hasil[i][j] = temp;
        }
    }
}
```



Solusi: coklat_2.cpp (lanj.)

```
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        int temp;
        scanf("%d", &temp);
        hasil[i][j] += temp;
    }
}

for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        printf("%d", hasil[i][j]);
        if (j+1 < N) {
            printf(" ");
        }
    }
    printf("\n");
}
}
```



Array Multidimensi

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

```
int data[2][50][50];
```

- Kita akan mendapatkan variabel $data[i][j][k]$ yang terdefinisi untuk $0 \leq i \leq 1$, dan $0 \leq j, k \leq 49$.



Catatan

- Pada saat array dideklarasikan, nilai yang ada di dalam array bisa jadi tidak tentu.
- Sebagai contoh, program berikut akan mencetak angka yang tidak tentu:

```
#include <stdio>

int main() {
    int arr[10];
    printf("%d\n", arr[0]);
}
```

- Pastikan Anda melakukan inisialisasi pada array dengan tepat.



Selanjutnya...

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

