

# TIF1101 – Dasar-Dasar Pemrograman

## HO 14 - Array

Opim Salim Sitompul

Department of Information Technology  
Universitas Sumatera Utara

# Outline

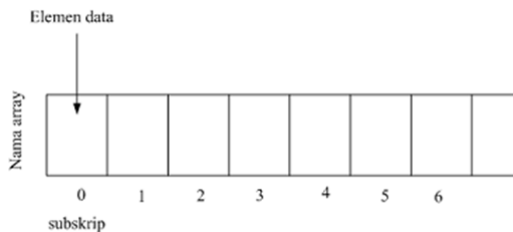
- 1 Pendahuluan
- 2 Array Satu Dimensi
  - Deklarasi Array Satu Dimensi
- 3 Mendefinisikan Ukuran Array
- 4 Inisialisasi Array Satu Dimensi
  - Memberikan harga awal pada saat deklarasi
  - Memberikan harga awal di dalam badan program
- 5 Array dan Pointer
- 6 Array sebagai Argumen Fungsi
- 7 Menyalin Elemen-elemen Array

# Pendahuluan

- Sekumpulan data yang sejenis dapat disimpan dalam sebuah objek data yang disebut array.
- Dengan dikumpulkannya data yang sejenis tersebut dalam sebuah objek data, pengacuan terhadap masing-masing objek data dapat dilakukan hanya dengan merujuk ke sebuah nama variabel.
- Hal ini sangat membantu sekali dalam hal efisiensi program, karena masing-masing elemen data tidak perlu memiliki nama variabel yang berbeda-beda.

# Array Satu Dimensi

- Secara diagramatis, konsepsi objek data array ini dapat digambarkan sebagai berikut:



Gambar 1: Objek Data Array



# Array Satu Dimensi

```
1 /* Nama file: rata2.c
2    Menghitung rata-rata bilangan */
3 #include <stdio.h>
4
5 int main()
6 {
7     /* Deklarasi array */
8     int bilangan[] = {6,12,11,5,27,10};
9     int i, jlh_bil, total = 0;
10    float rata2;
```



# Array Satu Dimensi

```
11     jlh_bil = sizeof(bilangan)/sizeof(int);
12     for(i=0;i<jlh_bil;i++)
13         total += bilangan[i];
14     rata2 = (float) total / (float) jlh_bil;
15     printf("Rata2 %d bilangan=%.2f\n", jlh_bil,
16           rata2);
17
17     return 0;
18 }
```

# Array Satu Dimensi

- Pengacuan elemen data tertentu di dalam array dapat dilakukan dengan menentukan posisi di mana elemen data itu terletak.
- Acuan letak elemen data ini disebut *subscript* atau indeks yang dimulai dari 0 (*zero subscripting*) dan diapit oleh sepasang kurung besar.
- Jumlah elemen data yang terdapat di dalam array ditentukan oleh sebuah bilangan yang disebut ukuran atau dimensi array.

# Deklarasi Array Satu Dimensi

- Bentuk umum pendeklarasian variabel array adalah:  
**jenis\_data** nama\_array[ukuran];
- Contoh:  
float tinggi[50];  
int jlh\_mhs[80];  
double berat[100];



# Deklarasi Array Satu Dimensi

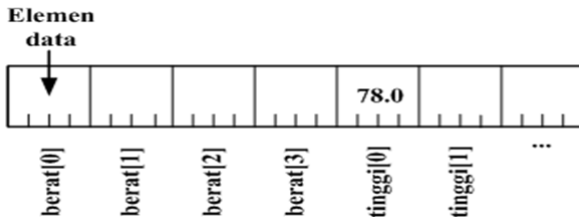
- Mendeklarasikan array artinya menyediakan sejumlah lokasi memori.
- Semua elemen array terletak di memori secara berurutan tanpa adanya ruang kosong di antara satu elemen dengan elemen berikutnya.
- Elemen array kedua diletakkan di sebelah elemen array pertama, elemen array ketiga di sebelah elemen array kedua, dan seterusnya.

# Deklarasi Array Satu Dimensi

- Untuk mengetahui jumlah byte yang disediakan oleh C untuk suatu array yang dideklarasikan:  
`n = sizeof>Nama_Array;`
- Dalam mengacu elemen-elemen array jangan melampaui ukuran array yang telah ditentukan.
- Perlu diperhatikan bahwa C tidak menyediakan fasilitas pemeriksaan batas (*error-bound checking*).

# Deklarasi Array Satu Dimensi

- Contoh:  
float berat[4];  
int tinggi[4];
- Kemudian diberikan nilai pada elemen array berat:  
berat[4] = 78.0;



# Mendefinisikan Ukuran Array

- Menggunakan prapengolah #define:  
#define MAXTINGGI 500  
#define MAXMHS 80  
#define MAXBERAT 100
- Atau menggunakan deklarasi konstanta:  
**const** int MAXTINGGI = 50;  
**const** int MAXMHS = 80;  
**const** int MAXBERAT = 100;

# Inisialisasi Array Satu Dimensi

- Sebelum digunakan, elemen-elemen array harus diberi harga awal.
  - Ada dua cara yang dapat dilakukan:
  - Memberikan harga awal pada saat deklarasi, dan
  - Memberikan harga awal di dalam badan program.

# Memberikan harga awal pada saat deklarasi

- Contoh:

```
int bilangan[ ] = {17,8,6,12,26,10};
```

- Ukuran array tidak diberikan
- Kompiler akan menghitung berapa jumlah elemen yang terdapat pada array

- atau

```
int bilangan[6] = {17,8,6,12,26,10};
```

- Pemberian ukuran array adalah bersifat opsional

- **Pemberian harga awal seperti berikut ini adalah salah:**

```
int bilangan[6];
```

```
...
```

```
/* Cara yang salah */
```

```
bilangan[6] = {6,12,11,5,27,10};
```

# Memberikan harga awal pada saat deklarasi

- Jika pada saat pendeklarasian diberikan harga awal 0 pada sebagian elemen pertama array, C akan memberikan harga awal pada elemen-elemen lainnya dengan nol.

- Contoh:

```
#define JLHBIL 20
```

```
...
```

```
long C[JLHBIL] = {1, 1, 0};
```

# Memberikan harga awal pada saat deklarasi

- Program berikut mengilustrasikan pemberian harga awal pada sebagian elemen array seperti dinyatakan di atas, sebuah program untuk mencetak daftar bilangan Catalan.
- Bilangan Catalan adalah bilangan-bilangan yang memenuhi aturan:
  - Dua buah bilangan Catalan yang pertama adalah 1, yaitu:  $C[0] = 1$  dan  $C[1] = 1$ .
  - Bilangan Catalan ke- $n$ , ditentukan dengan

$$C[n] = \sum_{i=0}^{n-1} (C[i] * C[n-i-1])$$





```
1 /* Nama file: catalan.c
2    Program untuk mencetak bilangan Catalan */
3 #include <stdio.h>
4 #define JLHBIL 20
5
6 void printArray(long []);
7
8 int main()
9 {
10     int i, j, n;
11     long C[JLHBIL] = {1, 1, 0};
12
13     printf("Harga awal array:\n");
14     printArray(C);
```

```
15 printf("Jumlah bilangan Catalan yang
    diinginkan: ");
16 scanf("%d", &n);
17 if (n<=JLHBIL)
18 {
19     printf("%d bil. Catalan pertama:\n",n);
20     printf("%ld %ld ", C[0], C[1]);
21     for(i=2; i<n; i++)
22     {
23         for(j=0; j<=i-1; j++)
24             C[i] = C[i] + (C[j] * C[i-j-1]);
25         printf("%ld ", C[i]);
26     }
```

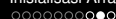


```
27     printf("\n");
28 }
29 else
30     printf("Bilangan terlalu besar, maksimum
        adalah %d...\n", JLHBIL);
31
32     return 0;
33 }
34
35 void printArray(long C[])
36 {
37     int i;
38
39     for(i=0; i <JLHBIL; i++)
40         printf("%d ", C[i]);
41     printf("\n");
42 }
```

# Memberikan harga awal di dalam badan program

- Menggunakan struktur pengulangan di dalam badan program
- Contoh:  

```
for(i=0; i<100; i++)  
    bilangan[i] = (rand() % 100)+1;
```



# Memberikan harga awal di dalam badan program

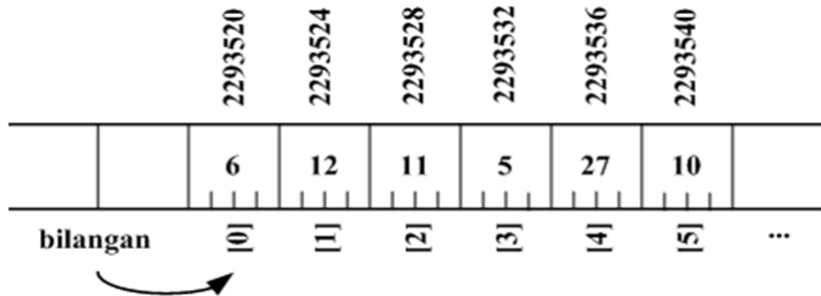
```
1  /* Namafile: arrayAcak
2     Menginisialisasi array dengan bilangan acak
3     */
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  #define N 100
9
10 int main()
11 {
12
13     int bilangan[N], i, totBil=0;
14     float rata2;
15
16     /* generate random array */
17     srand(time(NULL));
```

# Memberikan harga awal di dalam badan program

```
15  for(i=0; i<N; i++)
16  {
17      bilangan[i] = (rand() % N)+1;
18      totBil += bilangan[i];
19  }
20  rata2 = (float)totBil/(float)N;
21  /* print array */
22  for(i=0; i<N; i++)
23      printf("%d ", bilangan[i]);
24  printf("\n Total = %d\n", totBil);
25  printf(" Rata2 = %f\n", rata2);
26
27  return 0;
28 }
```

# Array dan Pointer

- Nama array merupakan sebuah pointer yang memuat alamat elemen array yang pertama (disebut *base address*).



# Array dan Pointer

- Metode pengacuan terhadap elemen-elemen array dapat dilakukan dengan cara lain.
- Dengan memanfaatkan faktor skala dari variabel pointer.
- Contoh:

(bilangan + 3)[0]    (3 + bilangan)[0]

(bilangan + 2)[1]    (2 + bilangan)[1]

(1 + bilangan)[2]    (bilangan + 1)[2]

(bilangan - 2)[5]    (-2 + bilangan)[5]



# Array dan Pointer

```
1 /* Namafile: elemenarr.c
2  Mendemonstrasikan elemen array */
3 #include <stdio.h>
4
5 int main()
6 {
7     int i, skala,
8         bilangan[6] = {6, 12, 11, 5, 27, 10};
9
10    printf("Elemen-elemen array:\n");
11    skala = 0;
12    for(i=0; i<6; i++)
13    {
14        printf("(bilangan + %d) --> ", skala);
15        printf("bilangan[%d] ", i);
```

# Array dan Pointer

```
16     printf("berisi nilai: %d\n", (bilangan +
17         skala) [0]);
18     skala++;
19 }
20 return 0;
21 }
```

# Array sebagai Argumen Fungsi

- Apabila array dikirim sebagai argumen ke sebuah fungsi, yang dikirim hanyalah alamat dasar (base address) dari array, tidak termasuk elemen-elemen array tersebut.
- Alamat dasar ini dikirim melalui pengiriman dengan nilai.
- Argumen formal yang terdapat pada definisi fungsi, bertindak sebagai sebuah pointer yang akan menerima alamat dasar array yang dikirim.

# Array sebagai Argumen Fungsi

```
1  /* Nama file: permacak.c
2     Program pembangkit permutasi acak */
3  #include <stdio.h>
4  #define N 10
5  #define TESTRUN 12
6
7  void cetakArray(int [], int);
8  void swap(int [], int, int);
9
10 int main()
11 {
12     int p[N];
13     int k, m, t, i, j;
```

## Array sebagai Argumen Fungsi

```
14  /* Inisialisasi array */
15  for(i = 0; i < N; i++)
16      p[i] = i;
17  printf("Permutasi Acak\n");
18  cetakArray(p, N);
19  /* Proses permutasi */
20  for(j = 0; j < TESTRUN; j++)
21  {
22      for(k = N-1; k > 0; k--)
23      {
24          m = rand() % k;
25          swap(p, k, m);
26      }
27      cetakArray(p, N);
28  }
29  return 0;
```

# Array sebagai Argumen Fungsi

```
31 void cetakArray(int p[], int n)
32 {
33     int i;
34
35     for(i=0; i < n; i++)
36         printf(" %2d", p[i]);
37     printf("\n");
38 }
39
40 void swap(int p[], int k, int m)
41 {
42     int t;
43
44     t = p[k];
45     p[k] = p[m];
46     p[m] = t;
47 }
```

# Array sebagai Argumen Fungsi

- Contoh:
  - Nama array sebagai pointer dikirimkan sebagai argumen fungsi.
  - Program Bubble Sort:

6	12	11	5	27	10
i				j-1	j

# Array sebagai Argumen Fungsi

```
1 /* Nama file: bubble.c
2     Mengurut bilangan dengan teknik bubble
3     sort */
4 #include <stdio.h>
5 #define N 6
6 void printArray(int []);
7 void bubbleSort(int []);
8
9 int main()
10 {
11     int a[] = {6, 12, 11, 5, 27, 10};
12
13     printf("Sebelum disortir:\n");
14     printArray(a);
```



# Array sebagai Argumen Fungsi

```
15  bubbleSort(a);
16  printf("Setelah disortir:\n");
17  printArray(a);
18
19  return 0;
20 }
21
22 void bubbleSort(int a[])
23 {
24     int i, j, temp;
25
26     for(i = 0; i < N-1; i++)
27         for(j = N-1; j > i; j--)
```

# Array sebagai Argumen Fungsi

```
29         if(a[j-1] > a[j])
30         {
31             temp = a[j - 1];
32             a[j - 1] = a[j];
33             a[j] = temp;
34         }
35     }
36
37 void printArray(int Array[])
38 {
39     int i;
40
41     for(i=0; i < N; i++)
42         printf("%d ", Array[i]);
43     printf("\n");
44 }
```

# Menyalin Elemen-elemen Array

- Operator penugasan (=) tidak dapat digunakan untuk memberikan nilai-nilai yang ada di sebuah array ke array yang lain.
- Contoh:  
`frekKuliah = nilaiMhs; /* tidak sah */`
- Menyalin elemen-elemen array ke array yang lain hanya dapat dilakukan dengan memindahkan masing-masing elemen tersebut satu persatu.
- Contoh:  
`for(i=0; i<100; i++)  
 frekKuliah[i] = nilaiMhs[i] ;`

# Menyalin Elemen-elemen Array

```
1 /* Nama file: hitungel.c
2  program frekwensi bilangan dalam array */
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <time.h>
6 #define JLHBIL 10
7
8 void inisialisasiArray(int [], int []);
9 void salinArray(int [], int []);
10 void printArray(int []);
11 void tampilkanFrekwensi(int [], int []);
```

# Menyalin Elemen-elemen Array

```
12 int main()
13 {
14     int bilangan[JLHBIL], salinan[JLHBIL];
15
16     inisialisasiArray(bilangan, salinan);
17     printArray(bilangan);
18     salinArray(bilangan, salinan);
19     tampilkanFrekwensi(bilangan, salinan);
20     printArray(salinan);
21
22     return 0;
23 }
```

## Menyalin Elemen-elemen Array

```
24 void salinArray(int bilangan[], int salinan[])
25 {
26     int i, j, sama;
27
28     /* Salin bilangan-bilangan yang berbeda */
29     for(i=0; i<JLHBIL; i++)
30     { /* cari apakah bilangan ke-i sudah ada di
31         array salinan */
32         sama = 0;
33         for(j=0; j<JLHBIL; j++)
34             if(salinan[j] == bilangan[i])
35                 sama = 1;
36         /* jika bilangan ke-i belum ada, bilangan
37            tersebut dimasukkan ke array salinan
38            pada posisi yang baru yg ditunjuk oleh
39            elemen -99 */
```

# Menyalin Elemen-elemen Array

```
40     if (!sama)
41     {
42         j = 0;
43         /* cari elemen yang masih kosong */
44         while(salinan[j] != -99) j++;
45         salinan[j] = bilangan[i];
46     }
47 }
48 }
49
50 void printArray(int Array[])
51 {
52     int i;
53
54     /* Tampilkan isi array */
```

# Menyalin Elemen-elemen Array

```
55 for(i=0; i<JLHBIL; i++)
56     printf("%d ", Array[i]);
57     printf("\n");
58 }
59
60 void tampilkanFrekwensi(int bilangan[], int
    salinan [])
61 {
62     int i, j, bil, ct;
63
64     /* Hitung bilangan yang berbeda */
65     bil = 0;
66     for(i=0; i<JLHBIL; i++)
67         if(salinan[i] != -99) bil++;
```



# Menyalin Elemen-elemen Array

```
69  /* Hitung frekwensi bilangan yang berbeda */
70  for(i=0; i<bil; i++)
71  {
72      ct = 0;
73      for(j=0; j<JLHBIL; j++)
74          if(salinan[i] == bilangan[j])
75              ct++;
76      printf("%d terjadi %d kali.\n", salinan[i], ct);
77  }
78 }
```