

Konsep Pemrograman

9. Array

Umi Sa'adah

Entin Martiana Kusumaningtyas

Tri Hadiah Muliawati

2020



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Overview

- Pendahuluan
- Array Berdimensi Satu
 - Mendeklarasikan Array
 - Mengakses Elemen Array
 - Menginisialisasi Array
 - Variasi dalam Mendeklarasikan Array
- Array Berdimensi Dua
 - Mendeklarasikan Array
 - Mengakses Elemen Array
- Array Berdimensi Banyak
- Inisialisasi Array Tak Berukuran
- Array Sebagai Parameter Fungsi



Pendahuluan

- Array adalah :
 - Sekumpulan data
 - Semua **tipe datanya HARUS sama**
 - Menggunakan **nama variabel yang sama**
 - Dibedakan **menurut indeksnya, by default dimulai dari 0 s/d (n-1)**, dengan n adalah jumlah elemen dalam array
- Array bisa berupa array berdimensi satu, dua, tiga atau lebih.
- Array berdimensi satu (*one-dimensional array*) mewakili bentuk suatu **vektor**.
- Array berdimensi dua (*two-dimensional array*) mewakili bentuk dari suatu **matriks atau tabel**
- Array berdimensi tiga (*three-dimensional array*) mewakili bentuk **suatu ruang**.



Mendeklarasikan Array

- Suatu array berdimensi satu dideklarasikan dalam bentuk umum berupa :

```
tipe_data nama_var[ukuran];
```

- dengan :
 - tipe_data : untuk menyatakan tipe dari elemen array, misalnya *int*, *char*, *float*.
 - nama_var : nama variabel array
 - ukuran : untuk menyatakan **jumlah maksimal** elemen array.
- Contoh pendeklarasian array :

```
float nilai[5];
```

menyatakan bahwa variabel `nilai` bertipe **array of float** dan memiliki 5 elemen bertipe *float*.

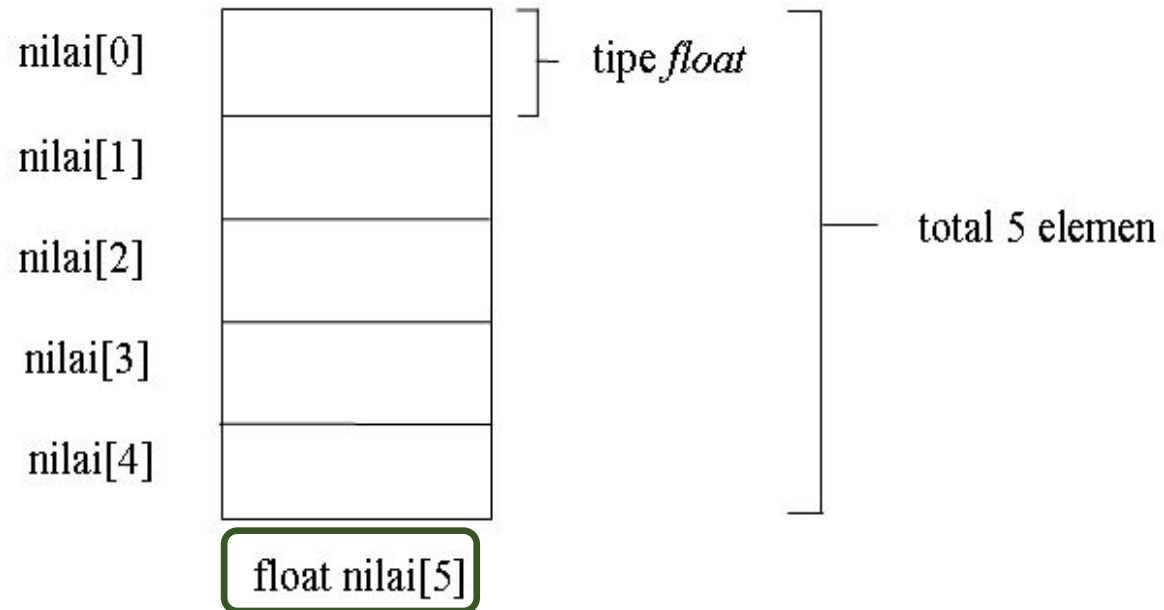


Mengakses Elemen Array

- Pada C, data array akan disimpan dalam memori yang berurutan
- Elemen pertama mempunyai indeks bernilai 0.
- Jika `nilai` dideklarasikan sebagai *array of float* dengan 5 elemen, maka elemen pertama memiliki indeks sama dengan 0, dan elemen terakhir memiliki indeks 4.



Mengakses Elemen Array



Bentuk umum pengaksesan array adalah sbb :

```
nama_var[indeks]
```

sehingga, untuk array `nilai`, maka :

`nilai[0]` → elemen pertama dari `nilai`

`nilai[4]` → elemen ke-5 dari `nilai`

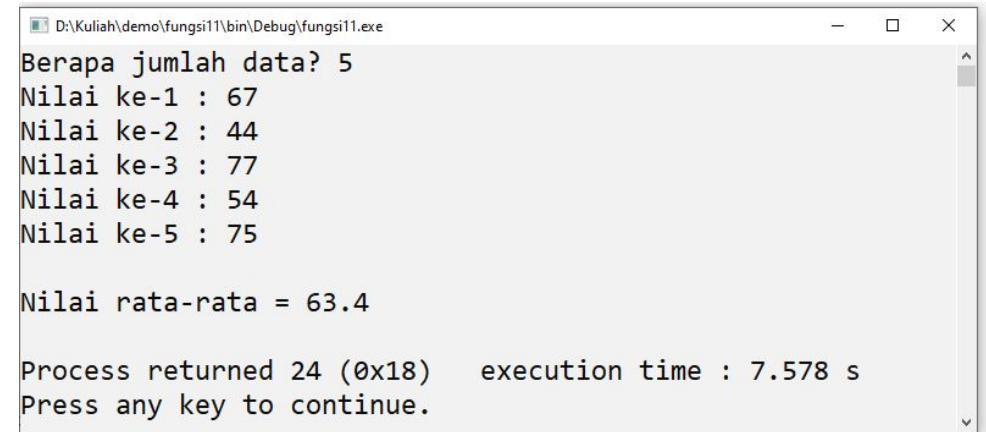
Contoh :

```
nilai[0] = 70;
scanf("%f", &nilai[2]);
```

Mengakses Elemen Array

```
#include <stdio.h>
#define MAKS 5
void main(){
    int i, n;
    float total=0, rata, nilai[MAKS];

    printf("Berapa jumlah data? ");
    scanf("%d", &n);
    for(i=0; i<n; i++) {
        fflush(stdin);
        printf("Nilai ke-%d : ", i+1);
        scanf("%f", &nilai[i]);
        total = total + nilai[i];
    }
    rata = total / n;
    printf("\nNilai rata-rata = %g\n", rata);
    return 0;
}
```



```
D:\Kuliah\demo\fungsi11\bin\Debug\fungsi11.exe
Berapa jumlah data? 5
Nilai ke-1 : 67
Nilai ke-2 : 44
Nilai ke-3 : 77
Nilai ke-4 : 54
Nilai ke-5 : 75

Nilai rata-rata = 63.4

Process returned 24 (0x18)   execution time : 7.578 s
Press any key to continue.
```

Inisialisasi Array

- Sebuah array dapat diinisialisasi sekaligus pada saat dideklarasikan.
- Untuk mendeklarasikan array, nilai-nilai yang diinisialisasikan dituliskan di antara kurung kurawal ({ }) yang dipisahkan dengan koma

Inisialisasi Array

```
#include <stdio.h>
void main() {
    int bln, thn, jhari;
    int jum_hari[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    puts("MEMPEROLEH JUMLAH HARI");
    puts("PADA SUATU BULAN DAN SUATU TAHUN");
    puts("-----");
    printf("Masukkan bulan (1..12)   : ");
    scanf("%d", &bln);
    printf("Masukkan tahunnya       : ");
    scanf("%d", &thn);
    if(bln == 2)
        if(thn % 4 == 0) //thn kabisat
            jhari = 29;
        else
            jhari = 28;
    else
        jhari = jum_hari[bln-1];
    printf("\nJumlah hari dalam bulan %d tahun %d adalah %d hari\n",bln,thn,jhari);
}
```

```
D:\Kuliah\demo\fungsi11\bin\Debug\fungsi11.exe
MEMPEROLEH JUMLAH HARI
PADA SUATU BULAN DAN SUATU TAHUN
-----
Masukkan bulan (1..12) : 2
Masukkan tahunnya : 2000

Jumlah hari dalam bulan 2 tahun 2000 adalah 29 hari

Process returned 53 (0x35) execution time : 7.473 s
Press any key to continue.
```



Variasi dalam Mendeklarasikan Array

- Ada beberapa variasi cara mendeklarasikan sebuah array (dalam hal ini yang berdimensi satu), di antaranya adalah sebagai berikut :

- `int numbers[10];`

- `int numbers[10] = {34, 27, 16 };`

- `int numbers[] = {2, -3, 45, 79, -14, 5, 0, 28, -1, 0 };`

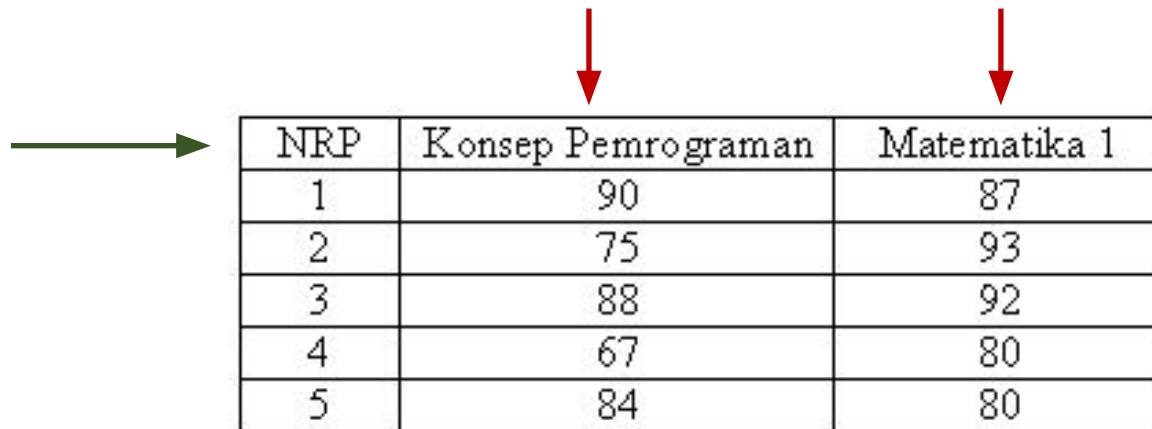
- `char text[] = "Welcome to New Zealand.";`

- `float radix[12] = {134.362f, 1913.248f};`

- `double radians[1000];`

- Pendeklarasian nama variabel array **diperbolehkan tidak disertai ukuran** yang mengindikasikan besarnya array asalkan langsung diinisialisasi.
- Dalam kondisi seperti ini, C akan menginisialisasi ukuran array tersebut sejumlah elemen yang diberikan di dalam kurung kurawal pada saat proses inisialisasi.

Array Berdimensi Dua



NRP	Konsep Pemrograman	Matematika 1
1	90	87
2	75	93
3	88	92
4	67	80
5	84	80

- Data seperti yang disajikan pada tabel di atas, dapat disimpan pada sebuah array berdimensi dua.
 - dimensi pertama dari array digunakan untuk menyatakan NRP
 - dimensi kedua untuk menyatakan Matakuliah

Mendeklarasikan Array (dimensi 2)

	0	1	
0	90	87	← indeks kedua (matakuliah)
1	75	93	
2	88	92	
3	67	80	
4	84	80	

↑ indeks pertama
(nrp)

`int nilai[5][2];`

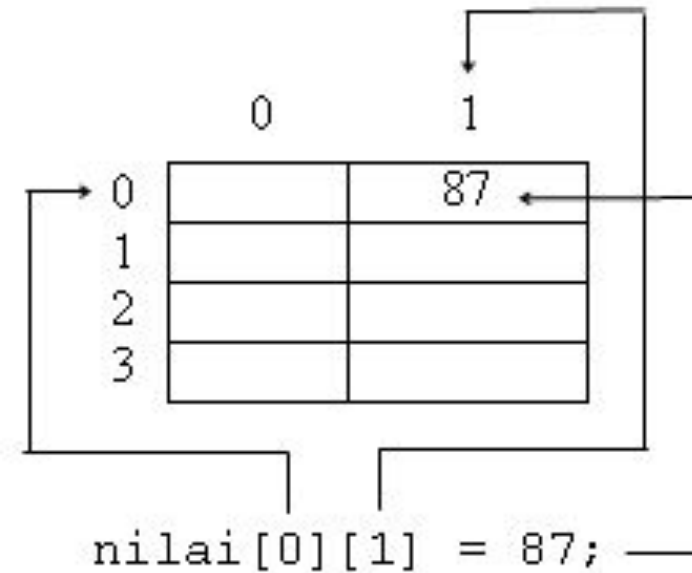
- Pendeklarasian yang diperlukan untuk menyimpan nilai mahasiswa pada tabel tsb adalah:

```
int nilai[5][2];
```

5 menyatakan banyaknya mahasiswa dan 2 menyatakan banyaknya matakuliah

Mengakses Elemen Array (dimensi 2)

- Array `nilai` dapat diakses dalam bentuk `nilai[indeks pertama][indeks kedua]`



•Contoh :

- `nilai[0][1] = 87;`
 - merupakan instruksi untuk memberikan nilai 87 ke array `nilai` untuk indeks pertama = 0 dan indeks kedua bernilai 1.
- `printf("%d", nilai[2][0]);`
 - merupakan perintah untuk menampilkan elemen yang memiliki indeks pertama = 2 dan indeks kedua = 0.

Array Berdimensi Banyak

- C memungkinkan untuk membuat array yang dimensinya lebih dari dua. Bentuk umum pendeklarasian array berdimensi banyak :
tipe nama_var[ukuran1][ukuran2}...[ukuranN];

- sebagai contoh :

```
int data_ruang[2][8][8];
```

merupakan pendeklarasian array **data_ruang** sebagai array berdimensi tiga.




Inisialisasi Array tak berukuran

- Inisialisasi array yang tak berukuran dapat dilakukan untuk array berdimensi satu atau lebih.
- Untuk array berdimensi lebih dari satu, dimensi terkirilah yang boleh tak berukuran.
- Dengan cara ini tabel dalam array dapat diperluas atau dikurangi tanpa mengubah ukuran array.
- Sebagai contoh :

```
int skala[] =  
    { 1, 2, 4, 6, 8 };
```

tak berukuran



merupakan pendeklarasian array berdimensi satu yang tak berukuran. Secara otomatis

skala[0] bernilai 1

skala[1] bernilai 2

skala[2] bernilai 4

skala[3] bernilai 6

skala[4] bernilai 8



Array sebagai Parameter Fungsi

Ketika diinginkan hasil proses di dalam fungsi terbaca di tempat fungsi tsb dipanggil, maka gunakan salah satu mekanisme sbb :

1. *return value* □ maks nilai yang dilaporkan = **SATU**
2. *pass by reference* □ nilai yg dilaporkan lebih dari 1, bisa berbeda tipe datanya
3. menjadikan array sbg parameter □ jika hasil prosesnya banyak dan semua tipenya sama.
 - jadikan array sebagai parameter aktual (tanpa kurung siku) & array sebagai parameter formalnya (tanpa size)
 - antara parameter aktual dengan parameter formal sebenarnya merupakan variabel yang berada pada lokasi/address yang SAMA, namun berbeda namanya (ALIAS)
 - SO, perubahan apapun pada parameter formal **PASTI** akan berpengaruh pada parameter aktual!!!



Array sebagai Parameter Fungsi

```
#define MAKS 20
```

```
void pemasukan_data(float []);
```

```
void pengurutan_data(float []);
```

```
void penampilan_data(float []);
```

```
int jml;
```

```
void main() {
```

```
    float data[MAKS];
```

```
    pemasukan_data(data);
```

```
    pengurutan_data(data);
```

```
    penampilan_data(data);
```

```
    return 0;
```

```
}
```

```
void pemasukan_data(float x[]) {  
    int i;  
  
    printf("Jumlah data = ");  
    scanf("%d", &jml);  
  
    for(i=0; i<jml; i++) {  
        printf("Data ke-%d : ", i+1);  
        scanf("%f", &x[i]);  
    }  
}
```

Array sebagai Parameter Fungsi

```

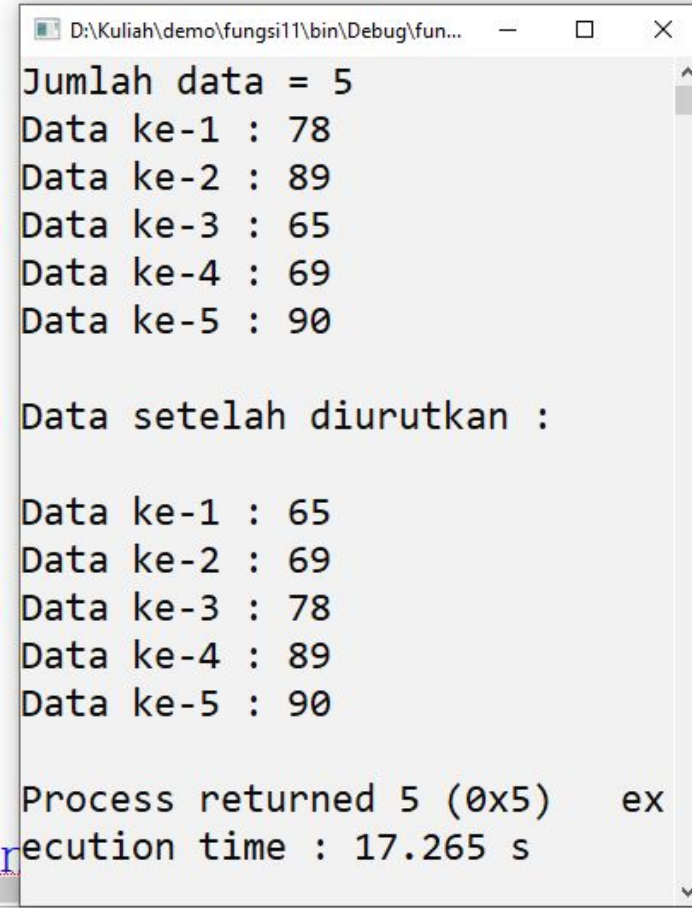
void pengurutan_data(float x[]) {
    int i, j;
    float smtr;

    for(i=0; i<jml-1; i++)
        for(j=i+1; j<jml; j++)
            if(x[i] > x[j]) {
                smtr = x[i];
                x[i] = x[j];
                x[j] = smtr;
            }
}

void penampilan_data(float x[]) {
    int i;

    printf("\nData setelah diurutkan :\n\n");
    for (i=0; i<jml; i++)
        printf("Data ke-%d : %g\n", i+1, x[i]);
}

```



```

D:\Kuliah\demo\fungsi11\bin\Debug\fun...
Jumlah data = 5
Data ke-1 : 78
Data ke-2 : 89
Data ke-3 : 65
Data ke-4 : 69
Data ke-5 : 90

Data setelah diurutkan :

Data ke-1 : 65
Data ke-2 : 69
Data ke-3 : 78
Data ke-4 : 89
Data ke-5 : 90

Process returned 5 (0x5)   ex
ecution time : 17.265 s

```

Latihan

1. Deklarasikan sebuah **variabel *array of int***, selanjutnya isi array tsb kemudian tampilkan isi variabel tersebut menggunakan statement **for ()**. Ingat apabila jumlah deklarasi *index* adalah **n** maka nilai *index*-nya adalah dimulai dari **0** sampai dengan **n-1**.
2. Deret fibonanci adalah deret yang dimulai dengan dua angka yang bernilai 0 dan 1, kemudian deret ketiga ditentukan dari penjumlahan kedua angka tersebut, sedangkan deret keempat ditentukan dari dua angka sebelumnya begitu seterusnya. Sehingga didapatkan deret fibonanci sebagai berikut: 0 1 1 2 3 5 8 13 21 ...

Buatlah program untuk meminta input dari user berupa sebuah bilangan, kemudian tampilkan deret fibonacci mulai dari 1 sampai dengan bilangan tsb.



Latihan

3. Buat penjumlahan dua matriks $A[2][2]$ dengan $B[2][2]$. Hasil penjumlahan tersebut disimpan dalam matrik C. Tampilkan di layar masing-masing isi dari matriks A, B dan C
4. Buatlah sebuah variabel *array of int* yang memiliki ukuran MAKS. Dengan menggunakan looping `for()`, masukkan sebanyak n data ke dalam array tsb kemudian carilah nilai terbesar yang ada dalam array tsb.



Latihan

5. Buatlah program yang terdiri atas :

- fungsi `input()` yang bertugas menanyakan jumlah data sekaligus memasukkan data bilangan sebanyak yang diminta ke dalam array `nilai[]`
- fungsi `findmax()` yang bertugas mencari nilai terbesar yang ada pada array `nilai[]` tsb
- fungsi `main()` yang memanggil fungsi `input()` dan fungsi `findmax()` , selanjutnya menampilkan nilai terbesar yang ada pada array `nilai[]`

Petunjuk : jadikan array sebagai parameter fungsi



Latihan

6. Diketahui daftar nilai siswa sebagai berikut:

No.	Nama Mhs	BAHASA	MATEMATIKA	DIGITAL
1.	Ahmad	81	90	62
2.	Adang	50	83	87
3.	Dani	89	55	65
4.	Edi	77	70	92

buatlah program untuk menampilkan laporan sebagai berikut:

No. Mhs	Rata-rata
1	77.67
2	73.33
3	69.67
4	79.67

Petunjuk: Gunakan variable array dua dimensi untuk menyimpan data.



Latihan

7. Buat program untuk menampilkan tabel berikut ini dengan menggunakan looping for dan 2 buah array :
- x sbg array berdimensi satu yang diinisialisasi dgn sudut 0 s/d 360
 - data sbg array berdimensi dua. data adalah array yg berisi nilai sin, cos dan tan dari masing-masing nilai sudut yang ada dalam array x .

Petunjuk :

- gunakan fungsi *radian()* untuk mengkonversi nilai derajat menjadi radian.

$$\text{rad} = \text{drjt} / 180.0 * \text{PI.}$$

$$\text{PI} = 3.14159$$

x	Sin(x)	Cos(x)	Tan(x)
0			
30			
60			
90			
120			
150			
180			
210			
240			
270			
300			
330			
360			



Referensi

1. Brian W. Kernighan, Dennis M. Ritchie (2012): The C Programming Language : Ansi C Version 2 Edition, PHI Learning
2. Byron Gottfried (2010) : Programming with C, Tata McGraw - Hill Education
3. Kochan Stephen (2004) : Programming in C, 3rd Edition, Sams
4. K. N. King (2008) : C Programming: A Modern Approach, 2nd Edition, W. W. Norton & Company
5. Abdul Kadir (2012) : Algoritma & Pemrograman Menggunakan C & C++, Andi Publisher, Yogyakarta
6. <http://www.gdsw.at/languages/c/programming-bbrowne/>
7. <https://www.petanikode.com/tutorial/c/>
8. <http://www.cprogramming.com/tutorial/c-tutorial.html>



bridge to the future

<http://www.eepis-its.edu>