

Pertemuan 11



Array, Array & Fungsi

Mata Kuliah : Algoritma & Pemrograman
Dosen : Tessy Badriyah, SKom., MT., PhD.



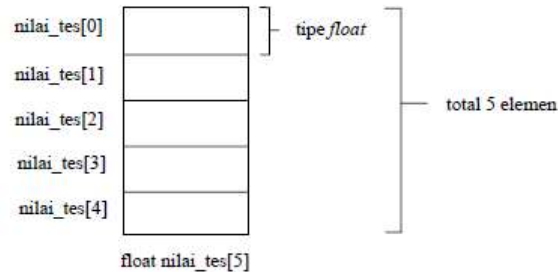
Tujuan Pembelajaran

- Memahami array berdimensi satu
- Memahami array berdimensi dua
- Memahami array berdimensi banyak
- Memahami inisialisasi array tak berukuran.
- Memahami array sebagai parameter fungsi



Array Berdimensi Satu

- Contoh pendeklarasian array :
 - `float nilai_tes[5];`
- Bentuk umum pengaksesan array
 - `nama_var[indeks]`





Variasi Deklarasi array dimensi satu

- `int bilangan[10];`
- `int bilangan[10] = { 34, 27, 16 };`
- `int bilangan [] = { 2, -3, 45, 79, -14, 5, 9, 28, -1, 0 };`
- `char teks[] = "Welcome to Indonesia.";`
- `float radix[12] = { 134.362, 1913.248 };`
- `double radian[1000];`



Program Array Berdimensi Satu

```
/* File program : nilai_tes.c
Pemakaian array utk menyimpan
sejumlah nilai tes */
#include <stdio.h>
#define MAKS 5
main()
{
int i;
float total = 0, rata;
float nilai_tes[MAKS]; /* deklarasi array
*/
for(i=0; i < MAKS; i++) /* pemasukan data
nilai_tes */
{
```

```
printf("Nilai tes ke-%d : ", i+1);
scanf("%f", &nilai_tes[i]);
/* menghitung jumlah seluruh nilai */
total = total + nilai_tes[i];
}
rata = total / MAKS; /* hitung nilai rata-
rata */
/* cetak nilai rata-rata */
printf("\nNilai rata-rata = %f\n", rata);
}
```



Program Array dimensi satu

```
/* File program : inisial.c */  
#include <stdio.h>  
main()  
{  
int i;  
int values[] = {1,2,3,4,5,6,7,8,9};  
char word[] = {'H','e','l','l','o'};  
for(i = 0; i < 9; ++i )  
printf("values[%d] is %d\n", i, values[i]);  
printf("\n");  
for(i = 0; i < 5; ++i )  
printf("word[%d] is %c\n", i, word[i]);  
}
```



Array berdimensi dua

- Data seperti yang disajikan pada Tabel berikut dapat disimpan pada sebuah array berdimensi dua.
- Dimensi pertama dari array digunakan untuk menyatakan kode program kursus dan dimensi kedua untuk menyatakan tahun kursus.

Data Kelulusan Siswa Pada Sebuah Kursus Komputer

Program	Tahun	1998	1999	2000
1. (INTRO)		80	540	1032
2. (BASIC)		15	83	301
3. (PASCAL)		8	12	15
4. (C)		10	129	257



Deklarasi array dimensi dua

- **Mendeklarasikan Array Berdimensi Dua**
 - `int data_lulus[4][3];`
- **Mengakses Elemen Array Berdimensi Dua**
 - `data_lulus[0][1] = 540;`

	0	1	2	← indeks kedua (tahun)
0	80	540	1032	
1	15	83	301	
2	8	12	15	
3	10	129	257	

↑ indeks pertama
(program kursus)

`int data_lulus[4][3];`




Program array dimensi banyak

```
/* File program : lulus.c
Contoh pemakaian array berdimensi dua */
#include <stdio.h>
main( )
{
int tahun, kode_program;
int data_lulus[4][3] ;
/* Memberikan data ke array */
data_lulus[0][0] = 80;
data_lulus[0][1] = 540;
data_lulus[0][2] = 1032;
data_lulus[1][0] = 15;
data_lulus[1][1] = 83;
data_lulus[1][2] = 301;
```



Program array dimensi banyak

```
data_lulus[2][0] = 8;
data_lulus[2][1] = 12;
data_lulus[2][2] = 15;
data_lulus[3][0] = 10;
data_lulus[3][1] = 129;
data_lulus[3][2] = 257;
/* proses utk memperoleh informasi jml siswa yg lulus */
printf("Masukkan tahun dr data yg ingin anda ketahui ");
printf("(1998..2000) : ");
scanf("%d", &tahun);
printf("Masukkan kode program kursus yang ingin anda
ketahui");
printf("(1 = INTRO, 2 = BASIC, 3 = PASCAL, 4 = C) : ");
scanf("%d", &kode_program);
printf("\nTotal kelulusan program tsb = %d\n",
data_lulus[kode_program - 1][tahun - 1998] );
}
```



Deklarasi dan Inisialisasi array dimensi dua

- Deklarasi dan inisialisasi yang dilakukan berupa :

```
int huruf_A[8][8] = {  
    { 0, 1, 1, 1, 1, 1, 0, 0 },  
    { 0, 1, 0, 0, 0, 1, 0, 0 },  
    { 0, 1, 0, 0, 0, 1, 0, 0 },  
    { 1, 1, 1, 1, 1, 1, 1, 0 },  
    { 1, 1, 0, 0, 0, 0, 1, 0 },  
    { 1, 1, 0, 0, 0, 0, 1, 0 },  
    { 1, 1, 0, 0, 0, 0, 1, 0 },  
    { 0, 0, 0, 0, 0, 0, 0, 0 }  
};
```



Program inisialisasi array dimensi dua

```
/* Contoh inisialisasi array dimensi dua */
#include <stdio.h>
main()
{
    int i,j;
    int huruf_A[8][8] = {
        { 0, 1, 1, 1, 1, 1, 0, 0 }, { 0, 1, 0, 0, 0, 1, 0, 0 }, { 0, 1, 0, 0, 0, 1, 0, 0 }, { 1, 1, 1, 1, 1, 1, 1, 0 },
        { 1, 1, 0, 0, 0, 0, 1, 0 }, { 1, 1, 0, 0, 0, 0, 1, 0 }, { 1, 1, 0, 0, 0, 0, 1, 0 }, { 0, 0, 0, 0, 0, 0, 0, 0 }
    };
    for(i = 0; i < 8; i++)
    {
        for(j = 0; j < 8; j++)
        if(huruf_A[i][j] !=0 )
            putchar('\xDB');
        else
            putchar(' '); /* spasi */
        putchar('\n');
    }
}
```



Array berdimensi banyak

- array yang dimensinya lebih dari dua
- Bentuk umum pendeklarasian array berdimensi banyak :
 - tipe nama_var[ukuran 1][ukuran2}...[ukuranN];
- contoh : `int data_huruf[2][8][8];`



Contoh inisialisasi array dimensi tiga

```
int data_huruf [2][8][8] =  
{ { { 0, 1, 1, 1, 1, 1, 0, 0 },  
  { 0, 1, 0, 0, 0, 1, 0, 0 },  
  { 0, 1, 0, 0, 0, 1, 0, 0 },  
  { 1, 1, 1, 1, 1, 1, 1, 0 },  
  { 1, 1, 0, 0, 0, 0, 1, 0 },  
  { 1, 1, 0, 0, 0, 0, 1, 0 },  
  { 1, 1, 0, 0, 0, 0, 1, 0 },  
  { 0, 0, 0, 0, 0, 0, 0, 0 }  
},  
{ { 1, 1, 1, 1, 1, 1, 0, 0 },  
  { 1, 0, 0, 0, 0, 1, 0, 0 },  
  { 1, 0, 0, 0, 0, 1, 0, 0 },  
  { 1, 1, 1, 1, 1, 1, 1, 0 },  
  { 1, 1, 0, 0, 0, 0, 1, 0 },  
  { 1, 1, 0, 0, 0, 0, 1, 0 },  
  { 1, 1, 1, 1, 1, 1, 1, 0 },  
  { 0, 0, 0, 0, 0, 0, 0, 0 }  
} };
```

- atau bisa juga ditulis menjadi



Contoh inisialisasi array dimensi tiga

- atau bisa juga ditulis menjadi

```
int data_huruf [2][8][8] =
```

```
0, 1, 1, 1, 1, 1, 0, 0,
```

```
0, 1, 0, 0, 0, 1, 0, 0,
```

```
0, 1, 0, 0, 0, 1, 0, 0,
```

```
1, 1, 1, 1, 1, 1, 1, 0,
```

```
1, 1, 0, 0, 0, 0, 1, 0,
```

```
1, 1, 0, 0, 0, 0, 1, 0,
```

```
1, 1, 0, 0, 0, 0, 1, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0,
```

```
1, 1, 1, 1, 1, 1, 0, 0,
```

```
1, 0, 0, 0, 0, 1, 0, 0,
```

```
1, 0, 0, 0, 0, 1, 0, 0,
```

```
1, 1, 1, 1, 1, 1, 1, 0,
```

```
1, 1, 0, 0, 0, 0, 1, 0,
```

```
1, 1, 0, 0, 0, 0, 1, 0,
```

```
1, 1, 1, 1, 1, 1, 1, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0,
```

```
};
```



Program array dimensi tiga

```
/* File program: data_huruf.c
Contoh inisialisasi array dimensi tiga */
#include <stdio.h>
main()
{
int i, j, k;
int data_huruf[2][8][8] = {
{{ 0, 1, 1, 1, 1, 1, 0, 0 },{ 0, 1, 0, 0, 0, 1, 0, 0 },{ 0, 1, 0, 0, 0, 1, 0, 0 },
{ 1, 1, 1, 1, 1, 1, 1, 0 },{ 1, 1, 0, 0, 0, 0, 1, 0 },{ 1, 1, 0, 0, 0, 0, 1, 0 },
{ 1, 1, 0, 0, 0, 0, 1, 0 },{ 0, 0, 0, 0, 0, 0, 0, 0 }
},
{{1, 1, 1, 1, 1, 1, 0, 0 },{1, 0, 0, 0, 0, 1, 0, 0 },{1, 0, 0, 0, 0, 1, 0, 0 },
{1, 1, 1, 1, 1, 1, 1, 0 },{1, 1, 0, 0, 0, 0, 1, 0 },{1, 1, 0, 0, 0, 0, 1, 0 },
{1, 1, 1, 1, 1, 1, 1, 0 },{0, 0, 0, 0, 0, 0, 0, 0 }
}
};
```




Program array dimensi tiga

```
for(i = 0; i < 2; i++) /* Tampilkan huruf */
{
    for(j = 0; j < 8; j++)
    {
        for(k = 0; k < 8; k++)
        if(data_huruf[i][j][k] != 0)
            putchar('\xDB');
        else
            putchar(' '); /* spasi */
        printf("\n"); /* pindah baris */
    }
    printf("\n"); /* pindah baris */
}
```



Inisialisasi array tak berukuran

- Inisialisasi array yang tak berukuran dapat dilakukan untuk array berdimensi satu atau lebih.
- Contoh :
 - `int skala[] = { 1, 2, 4, 6, 8 };`
 - `char konversi[][2] = { 'A', 'T', 'E', 'M', 'I', 'V', 'O', 'S', 'U', 'J', };`



Program array tak berukuran

```
/* File program : sandi.c
contoh inisialisasi array tak berukuran */
#include <stdio.h>
#include <stdlib.h>
#define JUM_KOLOM 2
#define MAX_KAR 256
main()
{
char konversi[][JUM_KOLOM] = { 'A', 'T', 'a', 't', 'E', 'M', 'e', 'm', 'I', 'V', 'i', 'v', 'O', 'S', 'o', 's',
'U', 'J',
'u', 'j'
};
char kalimat[MAX_KAR], karakter;
int i = 0, j, jum_kar, jum_penyandi;
printf("Masukkan sebuah kalimat dan akhiri dengan
ENTER\n");
printf("Kemudian kalimat tsb akan saya sandikan\n");
printf("Kalimat: ");
```



Program array tak berukuran

```
/* Memasukkan data karakter ke array kalimat */
while((kalimat[i] = getchar()) != '\n')
i++;
jum_kar = i ;
/* sandikan dan menampilkan ke layar */
printf("\nHasil setelah disandikan: ");
jum_penyandi = sizeof(konversi) / JUM_KOLOM;
for(i = 0; i < jum_kar; i++)
{
    karakter = kalimat [i];
    for(j = 0; j < jum_penyandi; j++)
    {
        if(karakter == konversi[j][0])
        {
            karakter = konversi[j][1];
            break; /* keluar dari for terdalam */
        }
    }
}
```



Program array tak berukuran

```
if(karakter == konversi[j][1])  
{  
    karakter = konversi[j][0];  
    break; /* keluar dari for terdalam */  
}  
}  
putchar (karakter);  
}  
printf("\n\n");  
}
```



Array sebagai parameter fungsi

- Array juga dapat dilewatkan sebagai parameter fungsi
- Sebagai digunakan untuk mengurutkan data
- Algoritma pengurutan
 1. Atur i bernilai 0
 2. Bandingkan $x[i]$ dengan $x[j]$, dengan j berjalan dari $i + 1$ sampai dengan $n-1$.
 3. Pada setiap perbandingan, jika $x[i] > x[j]$, maka isi $x[i]$ dan $x[j]$ ditukarkan
 4. Bila $i < (n - 1)$, ulangi mulai langkah 2.



Program Sorting dengan Array

```
/* File program : sorting.c */  
#include <stdio.h>  
#define MAKS 20  
void pemasukan_data(float [], int *);  
void pengurutan_data(float [], int);  
void penampilan_data(float [], int);  
main()  
{  
    float data[MAKS];  
    int jum_data;  
    pemasukan_data(data, &jum_data);  
    pengurutan_data(data, jum_data);  
    penampilan_data(data, jum_data);  
}
```



Program Sorting dengan Array

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




Program Sorting dengan Array

```
void pengurutan_data(float x[], int jumlah)
{
    int i, j;
    float smtr;
    for(i=0; i<jumlah-1; i++)
        for(j=i+1; j<jumlah; j++)
            if(x[i] > x[j]) /* penukaran data */
            {
                smtr = x[i];
                x[i] = x[j];
                x[j] = smtr;
            }
}
```



Program Sorting dengan Array

```
void penampilan_data(float x[], int jumlah)
{
    int i;
    printf("\nData setelah diurutkan :\n\n");
    for (i=0; i<jumlah; i++)
        printf("Data ke-%d : %g\n", i+1, x[i]);
}
```



Yang sudah dipelajari

- Telah dipelajari tentang Array dan contoh program yang menggunakan Array
- Penggunaan Array pada Fungsi



Referensi

- Robertson, Lesley Anne. (1992). *Students' guide to program design*. Oxford : Newnes
- Santner, Williams, and Notz (2003), *Design and Analysis of Computer Experiments*, Springer.
- Deitel & Deitel, *C How to Program*, Prentice Hall 1994 (2nd edition)
- Brookshear, J.G., *Computer Science: An Overview*, Benjamin-Cummings 2000 (6th edition)
- Kernighan & Ritchie, *The C Programming Language*, Prentice Hall