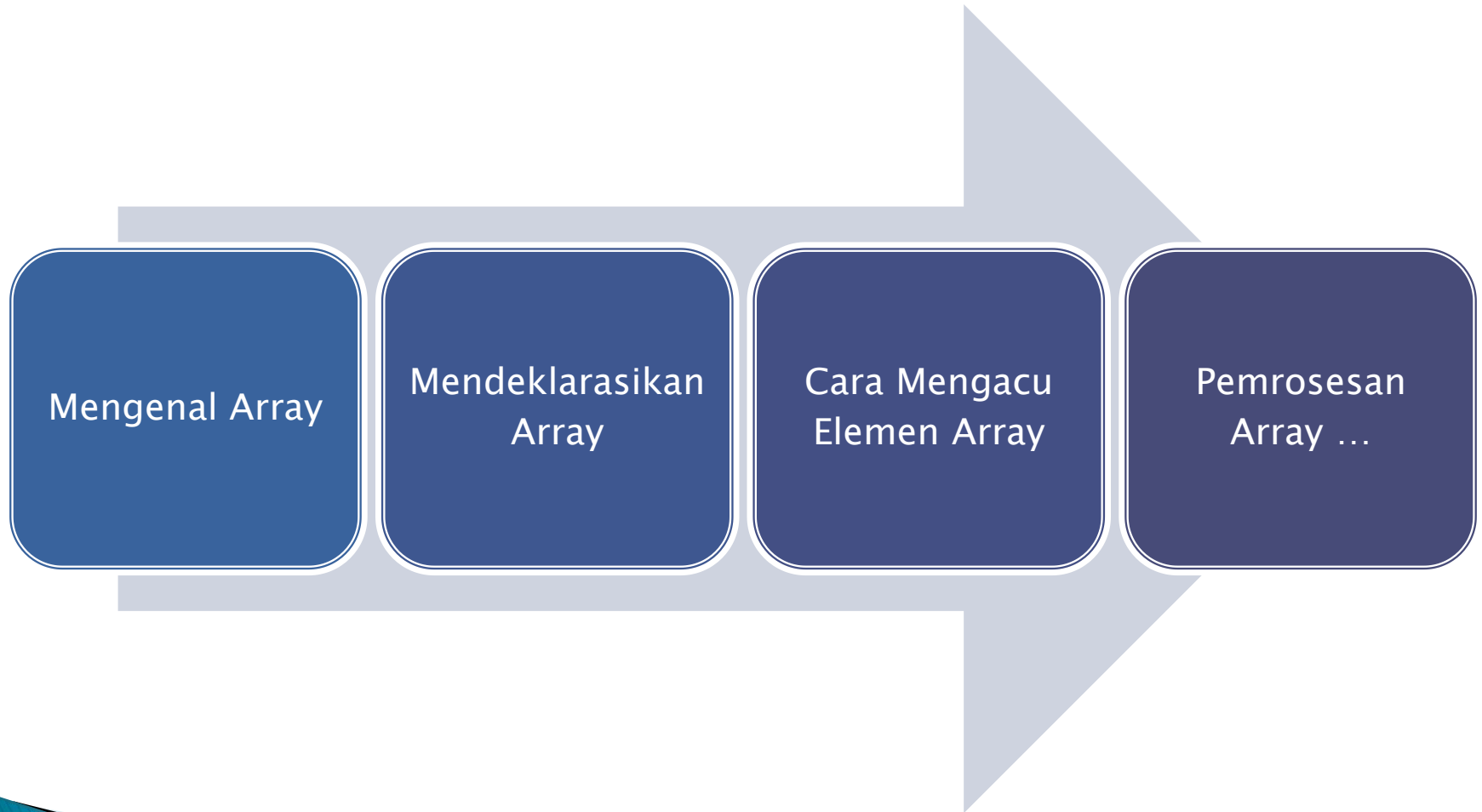


Array

(bagian 2)

Materi Sebelumnya...




Materi Sebelumnya...

... Pemrosesan Array

- Ukuran Efektif Array
- Menginisialisasi Array
- Mengisi elemen array dengan pembacaan
- Mencetak elemen-elemen array
- Menghitung nilai rata-rata
- Mencari nilai maksimum
- Mencari nilai minimum
- Menyalin array
- Menguji kesamaan dua buah array

Pelajaran dari Materi Sebelumnya

- ▶ **Array** adalah struktur data yang menyimpan sekumpulan elemen yang **bertipe sama**.
 - ▶ Setiap elemen diakses langsung melalui indeksnya.
 - ▶ Array adalah **struktur yang statis**, artinya jumlah elemen array harus sudah diketahui sebelum program dieksekusi.
- 

Selanjutnya ...

- Array bertipe terstruktur
- Bekerja dengan dua buah array
- String sebagai array karakter

Array Bertipe Terstruktur

- ▶ Misalkan kita akan mengolah data 31 orang mahasiswa.
- ▶ Data setiap mahasiswa terdiri atas NPM, nama mahasiswa, dan IPK (nilainya berkisar antara 0..4)

	NIM	NamaMhs	IPK
1	1402007021	Wisnu Guritno	3.15
2	1402009003	Andri Ilham	3.25
:			
:			
31	1402011026	Siti Maria Ulfah	3.30

- ▶ Struktur array yang menyimpan data mahasiswa tersebut dideklarasikan di bawah ini:

DEKLARASI

```
const Nmaks = 32
type Mahasiswa : record
    <NPM      : integer,
      NamaMhs : string,
      IPK     : real
    >
type TabMhs : array[1..Nmaks]
              of Mahasiswa
Mhs : TabMhs
```

Pendefinisian tipe data terstruktur dengan nama Mahasiswa

Notasi Algoritmik

```
#define Nmaks 32
typedef char String[25];
typedef struct{
    int    NPM;
    String NamaMhs;
    float  IPK;
}Mahasiswa;
typedef Mahasiswa TabMhs[Nmaks];
```

Pendefinisian tipe data String berupa array karakter

Pendefinisian tipe data TabMhs berupa array of mahasiswa

```
TabMhs Mhs;
```

Pendeklarasian variable Mhs dengan tipe data TabMhs

Bahasa C

Cara Mengacu Elemen

► Cara Mengacu Elemen Mhs

<code>Mhs[2]</code>	{elemen kedua dari array <code>Mhs</code> }
<code>Mhs[2].NPM</code>	{mengacu field NIM dari elemen kedua array}
<code>Mhs[2].IPK</code>	{mengacu field IPK dari elemen kedua array}

- ▶ Karena record merupakan struktur, maka kita tidak dapat melakukan pencetakan elemen array Mhs seperti di bawah ini:

```
write (Mhs[i])
```

- ▶ Tetapi haruslah seperti berikut:

```
write (Mhs[i].NPM, Mhs[i].NamaMhs, Mhs[i].IPK)
```

- ▶ Namun cara mengisi nilai elemen ke elemen lainnya dengan penulisan seperti di bawah ini adalah benar:

```
Mhs[i] ← Mhs[i+1]
```

- ▶ Yang sama saja dengan :

```
Mhs[i].NPM ← Mhs[i+1].NPM
```

```
Mhs[i].NamaMhs ← Mhs[i+1].NamaMhs
```

```
Mhs[i].IPK ← Mhs[i+1].IPK
```

Algoritma Mengisi Array Mhs

procedure BacaDataMahasiswa(input n : integer, output
Mhs : TabMhs)

{Membaca data mahasiswa (NIM, nama, IPK)}

{K. Awal : n berisi jumlah data mahasiswa}

{K. Akhir : Mhs berisi data hasil pembacaan}

DEKLARASI

i : integer

ALGORITMA:

for i \leftarrow 1 to N do

read(Mhs[i].NIM)

read(Mhs[i].NamaMhs)

read(Mhs[i].IPK)

endfor

Translasi Notasi Algoritmik Array ke dalam Bahasa C



Array sebagai Parameter Masukan

```
void CetakArray(ArrayInt A, int n)
/* Mencetak elemen-elemen array [1..n] */
```

Cara pemanggilan : CetakArray(A, n)

Prototipe prosedur di atas ekivalen dengan penulisan berikut:

```
void CetakArray(int A[], int n)
/* Mencetak elemen-elemen array [1..n] */
/* ukuran array boleh tidak perlu karena diset di program utama */
```

atau ekivalen dengan:

```
void CetakArray(int *A, int n)
/* Mencetak elemen-elemen array [1..n] */
```

Catatan:

Array dalam bahasa C adalah pointer, sehingga dapat ditulis **int** *A saja.

Array sebagai Parameter Keluaran

```
void BacaArray(ArrayInt A, int n);  
/* Mengisi elemen-elemen array A[1..n] */
```

Cara pemanggilan : `BacaArray(A, n)`

Catatan :

Karena array dalam bahasa C adalah pointer, maka tidak diperlukan karakter “*” di awal nama parameter tersebut.

Array sebagai Parameter Masukan/Keluaran

```
void UrutArray(ArrayInt A, int n);  
/* Mengurut elemen-elemen array A[1..n] sehingga  
tersusun menaik */
```

Cara pemanggilan : `UrutArray(A, n)`

Catatan :

Karena array dalam bahasa C adalah pointer, maka tidak diperlukan karakter “*” di awal nama parameter tersebut.

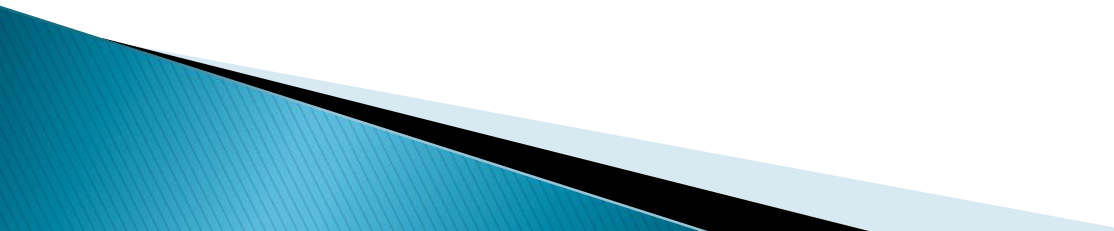
Selanjutnya...

»» Array of Array

Array of Array

- ▶ Selain bertipe terstruktur, elemen array juga dapat bertipe array lain.
- ▶ Misalnya kita ingin mengolah data nilai semua mata kuliah yang diambil setiap mahasiswa di semester genap.
- ▶ Asumsikan setiap mahasiswa mengambil 4 buah mata kuliah yang berbeda.
- ▶ Setiap elemen array berisi data sebagai berikut :
 - NPM
 - NamaMhs
 - Mata kuliah (MK) yang diambil mahasiswa tersebut (4 buah) berupa array.

► Mata Kuliah berupa array:

- Kode mata kuliah ke-1
Nama mata kuliah ke-1
Nilai mata kuliah ke-1
 - Kode mata kuliah ke-2
Nama mata kuliah ke-2
Nilai mata kuliah ke-2
 - Kode mata kuliah ke-3
Nama mata kuliah ke-3
Nilai mata kuliah ke-3
 - Kode mata kuliah ke-4
Nama mata kuliah ke-4
Nilai mata kuliah ke-4
- 

Struktur array yang menyimpan data tersebut dideklarasikan di bawah ini.

Nama array adalah Mhs2

DEKLARASI

```
const Nmaks = 100
```

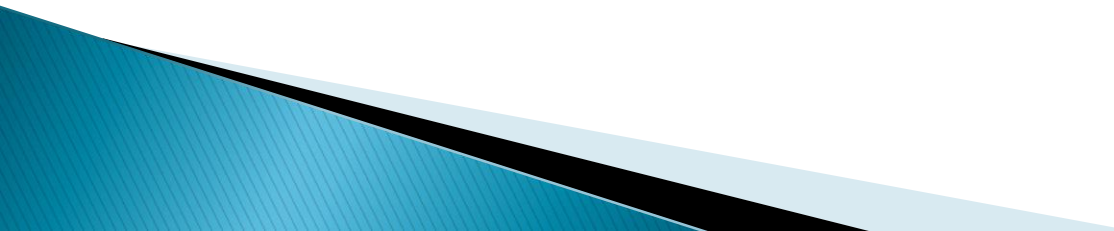
```
type MataKuliah : record <KodeMK   : string,  
                           Nilai    : char  
                           >
```

```
type Mahasiswa   : record <NIM      : integer,  
                           NamaMhs  : string,  
                           MK       : array[1..4] of MataKuliah  
                           >
```

```
type TabMhs : array[1..Nmaks] of Mahasiswa
```

```
Mhs2 : TabMhs
```

Bekerja dengan dua buah Array

- ▶ Dua buah array atau lebih dapat dioperasikan sekaligus.
 - ▶ Misalkan nilai ujian n orang mahasiswa telah disimpan di dalam array NilUjian.
 - ▶ Kita akan menghitung nilai indeks (A/B/C/D/E) mahasiswa tersebut dan menyimpan nilai indeks tersebut di dalam array Indeks.
- 

► Indeks nilai ujian ditentukan sebagai berikut:

$\text{NilUjian}(k) \geq 80$, indeks nilai = A
$70 \leq \text{NilUjian}(k) < 80$, indeks nilai = B
$55 \leq \text{NilUjian}(k) < 70$, indeks nilai = C
$45 \leq \text{NilUjian}(k) < 55$, indeks nilai = D
$\text{NilUjian}(k) < 45$, indeks nilai = E

► Deklarasi data:

DEKLARASI

const Nmaks = 200

type ArrayUjian : array[1..Nmaks] of real

type ArrayIndeks : array[1..Nmaks] of char

Prosedur menghitung nilai indeks

```
procedure HitungIndeksNilai(input NilUjian : ArrayUjian, input n : integer, output
Indeks : Array Indeks)
    {Menghitung indeks nilai ujian n orang mahasiswa}
    {K. Awal : n sudah berisi jumlah mahasiswa; elemen array NilUjian[1..n] sudah terdefinisi
    harganya}
    {K. Akhir : array Indeks [1..n] berisi nilai indeks ujian}
```

DEKLARASI

```
    i : integer      {indeks array}
```

DESKRIPSI

```
    for k  $\leftarrow$  1 to n do
        if NilUjian[i]  $\geq$  80 then
            Indeks[i]  $\leftarrow$  'A'
        else
            if (NilUjian  $\geq$  70) and (NilUjian[i] < 80) then
                Indeks[i]  $\leftarrow$  'B'
            else
                if (NilUjian  $\geq$  55) and (NilUjian[i] < 70) then
                    Indeks[i]  $\leftarrow$  'C'
                else
                    if (NilUjian  $\geq$  45) and (NilUjian[i] < 55) then
                        Indeks[i]  $\leftarrow$  'D'
                    else Indeks[i]  $\leftarrow$  'E'
                    endif
                endif
            endif
        endif
    endif
```

Prosedur mencetak tabel array

```
procedure CetakNilai(input NilUjian : ArrayUjian, input Indeks :  
ArrayIndeks, input n : integer)  
{mencetak tabel yang berisi nilai ujian dan indeks nilainya. Nomor peserta  
ujian sesuai dengan nomor indeks array}  
{K. Awal : NilUjian[1..N] dan Indeks[1..n] sudah terdefinisi elemen-  
elemennya}  
{K. Akhir : nilai ujian dan nilai indeksnya tercetak}
```

DEKLARASI

```
    i : integer {indeks array}
```

ALGORITMA

```
{cetak header table}  
write('-----')  
write(' No      Nilai Ujian                Indeks      ')  
write('-----')  
{cetak nilai ujian dan indeks mahasiswa ke-i}  
for i  $\leftarrow$  1 to N do  
write(i, '      ', NilUjian[i], '                ', Indeks[i])  
{cetak garis penutup}  
write('-----')
```

String sebagai Array Karakter

- ▶ String pada hakikatnya adalah array karakter dengan panjang dinamis, artinya ukuran array baru ditentukan pada saat program *running*.
- ▶ Karena string adalah array, maka elemen–elemen string yang berupa karakter dapat diakses melalui indeks.
- ▶ Contoh, jika s adalah peubah bertipe string, dan s berisi konstanta string : 'ini string'
- ▶ Maka dapat dikatakan bahwa s panjangnya 10 karakter dan elemen–elemennya adalah:

```
s[1] = 'i'    s[2] = 'n'    s[3] = 'i'    s[4] = ' '  
s[5] = 's'    s[6] = 't'    s[7] = 'r'    s[8] = 'i'  
s[9] = 'n '   s[10] = 'g'
```

- Dalam bahasa C, peubah bertipe string dideklarasikan dengan array bertipe char beserta panjangnya, seperti contoh berikut :

```
char String[20];
```

- Dalam Bahasa C, array karakter diakhiri dengan karakter null '\0'

```
'ini string'
```

string

- | | | | | | | | | | | |
|---|---|---|--|---|---|---|---|---|---|----|
| i | n | i | | s | t | r | i | n | g | \0 |
|---|---|---|--|---|---|---|---|---|---|----|

- Indeks string adalah dari 0 sampai n. n adalah panjang string.
- Dalam bahasa C, string kosong adalah array yang panjangnya 0 dan hanya diisi dengan karakter '\0'

Menghitung Panjang String

- ▶ Panjang string dihitung dengan memindai (scan) elemen-elemen array karakter sampai ditemukan karakter '\0'

```
int Panjang(char s[])
/* mengembalikan panjang string */
{
    int i;
    i = 0;
    while (s[i] != '\0')
        i = i + 1;
    /* end while */
    /* s[i] = '\0' */
    return i;
}
```

Menyalin string

```
void SalinString(char s1[], char s2[])
/* menyalin string s1, menghasilkan string s2
   K. Awal : s1 sudah terdefinisi
   K. Akhir : s2 adalah salinan s1, termasuk '\0'
*/
{
    int i;
    i = 0;
    while (s1[i] != '\0')
    {
        s2[i] = s1[i];
        i = i + 1;
    }
    /* end while */
    /* s1[i] = '\0' */
    s2[i] = '\0';
}
```

Fungsi Standard C untuk Salin String

- ▶ Bahasa C sudah menyediakan fungsi standard untuk menyalin string, yaitu *strcpy*. Fungsi ini didefinisikan dalam file header `<string.h>`

```
strcpy(t, s); /* copy s ke t */
```

Membandingkan dua buah string

```
boolean ApakahSama(char s1[], char s2[])
// 1 jika s1==s2, atau 0 jika s1 != s2
{
    int i;
    boolean sama;
    sama = TRUE; // asumsikan sementara bahwa s1 dan s2 sama

    // cek panjang string
    printf("panjang s1 = %d\n", strlen(s1));
    printf("panjang s2 = %d\n", strlen(s2));
    if (strlen(s1) != strlen(s2)) // s1 dan s2 tidak sama panjangnya
        sama = FALSE;
    else
        // panjang s1 dan s2 sama, cek kesamaan elemen-elemennya
        {
            i = 1;
            while (s1[i] != '\0' && sama)
                if (s1[i] == s2[i])
                    i=i+1;
                else // s1[i] != s2[i], maka dapat disimpulkan s1!=s2
                    sama = FALSE;
            //end if
        }
    //end while
    //s[i] = '\0' || not sama
}
return sama;
}
```

Program Mahasiswa



Deklarasi...

```
#include <stdio.h>
#define Nmaks 50

typedef char String[25];

typedef struct{
    String    KodeMK;
    int       Nilai;
}MataKuliah;

typedef struct{
    int       NPM;
    String     NamaMhs;
    MataKuliah MK[4];
}Mahasiswa;

typedef Mahasiswa TabMhs[Nmaks];

void InisialisasiArray(int n, TabMhs Mhs);
void BacaDataMahasiswa(int n, TabMhs Mhs);
void TulisDataMahasiswa(int n, TabMhs Mhs);
```

... Program Utama ...

```
main()
{
    TabMhs Mhs;
    int n;
    printf("Masukkan jumlah mahasiswa : ");
    scanf("%d",&n);
    InisialisasiArray(n, Mhs);
    BacaDataMahasiswa(n, Mhs);
    TulisDataMahasiswa(n, Mhs);
    system("PAUSE");
}
```

... Prosedur Inisialisasi Array ...

```
void InisialisasiArray(int n, TabMhs Mhs)
{
    /* Menginisialisasi array dengan nilai tertentu
       K. Awal : n berisi jumlah data mahasiswa
       K. Akhir : Array Mahasiswa sudah diinisialisasi */

    int i, j;
    for (i=1; i<=n;i++)
    {
        Mhs[i].NPM          = 0;
        Mhs[i].NamaMhs[i] = ' ';
        for (j=0; j<=4; j++)
        {
            Mhs[i].MK[j].KodeMK[j] = ' ';
            Mhs[i].MK[j].Nilai      = 0;
        }
    }
}
```


... Prosedur BacaDataMahasiswa...

```
void BacaDataMahasiswa(int n, TabMhs Mhs)
{
/* Membaca data mahasiswa (NIM, nama, IPK)
   K. Awal : n berisi jumlah data mahasiswa
   K. Akhir : Mhs berisi data hasil pembacaan */

int i,j;
for (i=1;i<=n;i++)
    {
        printf("Data Mahasiswa ke-%d \n", i);
        printf("Masukkan NPM Mahasiswa  :"); scanf("%d",&Mhs[i].NPM);
        printf("Masukkan Nama Mahasiswa :"); scanf("%s",&Mhs[i].NamaMhs);
        for (j=1;j<=4;j++)
        {
            printf("Mata Kuliah ke-%d \n",j);
            printf("Nama Matakuliah :"); scanf("%s",&Mhs[i].MK[j].KodeMK);
            printf("Nilai          :"); scanf("%d",&Mhs[i].MK[j].Nilai);
        }
    }
}
```

... Prosedur TulisDataMahasiswa

```
void TulisDataMahasiswa(int n, TabMhs Mhs)

{
/* Membaca data mahasiswa (NIM, nama, IPK)
   K. Awal : n berisi jumlah data mahasiswa
   K. Akhir : Mhs berisi data hasil pembacaan */

int i,j;
for (i=1;i<=n;i++)
    {
        printf("Data Mahasiswa ke-%d \n", i);
        printf("NPM Mahasiswa : %d \n",    Mhs[i].NPM);
        printf("Nama Mahasiswa : %s \n",    Mhs[i].NamaMhs);
        for (j=1;j<=4;j++)
        {
            printf("Mata Kuliah ke-%d : \n",j);
            printf("Nama Matakuliah : %s \n", Mhs[i].MK[j].KodeMK);
            printf("Nilai                : %d \n", Mhs[i].MK[j].Nilai);
        }
    }
}
```

Referensi

- ▶ Munir, Rinaldi. (2011). Algoritma & Pemrograman dalam Bahasa Pascal dan C. Edisi Revisi. Penerbit Informatika : Bandung.
 - Bab 12 : Larik