



**UNIVERSITAS
ATMA JAYA YOGYAKARTA**
serviens in lumine veritatis



Dasar Pemrograman (INFT06204)

Minggu 11
Array





Array

- **Variable**

- Menyimpan sebuah nilai dari tipe data tertentu.

- **Array**

- Menyimpan sekumpulan nilai yang tipe data tertentu (sama).

- Dalam bidang ilmiah sering ditemui besaran yang menggunakan nama peubah berindeks.

Misalnya :

- $u_1, u_2, u_3, u_4, u_5, \dots, u_n$.
- $v_k \geq 0$, untuk $k = 0, 1, 2, \dots, n$



Array (2)

- Bila kumpulan data tersebut disimpan secara beruntun di dalam memori, maka tiap elemen dapat diacu dengan menggunakan indeks.
- **Indeks** menyatakan posisi data relatif di dalam kumpulan.
- Struktur data penyimpanan data seperti ini dinamakan **larik (*array*)**.
- Nama lain larik : **tabel**, **vektor**, atau **peubah majemuk**.



Definisi Array

- **Array** adalah struktur data yang menyimpan sekumpulan elemen yang bertipe sama, setiap elemen diakses secara langsung melalui indeksnya.
- Indeks array haruslah tipe data yang menyatakan **keterurutan**, misalnya : integer, atau karakter.
- Contoh :
 - Larik yang bernama A dengan 8 buah elemen
→

A	
0	100
1	125
2	140
3	120
4	110
5	156
6	142
7	180

Tipe data harus sama

Index



Definisi Array (2)

- Notasi tiap elemen :
 $A[0], A[1], A[2], A[3], A[4], A[5], A[6], A[7]$
- Angka di dalam kurung siku menyatakan indeks.
- Setiap elemen menyimpan sebuah nilai.



Mendefinisi Array di dalam DEKLARASI

- Array adalah struktur data yang **statik**.
- Jumlah elemen array harus sudah diketahui sebelum program dieksekusi.
 - *tidak dapat ditambah atau dikurangi*
- Mendefinisikan array di dalam DEKLARASI:
 - Mendefinisikan **banyaknya elemen array**
 - Memesan sejumlah tempat di memori (sebanyak elemen array).
 - Mendefinisikan **tipe elemen array**
 - Tipe sederhana: integer, real, char, boolean
 - Tipe bentukan : record, string
 - Tipe larik



Contoh

- Array sebagai VARIABLE
 - `float data[10];`
 - `int A[5] = {1,2,3,4,5};`
 - `char kar[] = {'A','B','C'};` atau `char kar[] = "ABC";`
- Array sebagai TIPE BARU
 - `typedef int data[10];`
`data nilai;`
- Ukuran maksimum array sebagai konstanta
 - `#define N 10`
`typedef int data[N];`
`data nilai;`



Cara Mengacu Elemen Array

- Elemen array diacu melalui indeksnya.
- Misal :
 - `L[4] = 10;`
 - `nama_mhs[i] = "linuxita";`
 - `scanf("%d", &P[k]);`
 - `printf("%d", deret[n]);`



Pemrosesan Array

- Elemen array tersusun secara beruntun
→ elemen array diproses secara beruntun melalui indeks yang terurut.
- Skema umum algoritma memproses array disebut juga skema mengunjungi (***traversal***) array.

```
#define N 5
typedef int DataKu[N];

int main()
{
    DataKu data;
    int i;

    for(i=1;i<=N;i++)
    {
        printf("Data-%d: ",i); scanf("%d", &data[i]);
    }
    :
}
```

- → PROSES adalah aksi yang dilakukan terhadap elemen array (*pengisian, pembacaan, penulisan, atau manipulasi lainnya*)



Menginisialisasi Array

- Menginisialisasi array adalah memberikan harga awal untuk seluruh elemen larik.
 - Untuk ‘mengosongkan’ elemen array sebelum digunakan untuk proses tertentu.
 - Array bertipe numerik biasanya diisi dengan bilangan nol.
 - Array bertipe karakter biasanya diisi dengan karakter spasi.
 - Pemrogram dapat memberi nilai lain sesuai dengan kebutuhan.



Memanipulasi Array

- Seringkali tidak seluruh elemen array (yang telah didefinisikan) digunakan.
- Banyaknya elemen array yang digunakan disebut **jumlah elemen efektif**.

```
#define N 50

typedef int DataKu[N];

int main()
{
    DataKu data;
    int i, nEf;

    printf("Banyak data: "); scanf("%d", &nEf);
    for(i=1; i<=nEf; i++)
    {
        printf("Data-%d: ",i); scanf("%d", &data[i]);
    }
}
```



Kapan Menggunakan Array ?

- Array dibutuhkan bila diperlukan penyimpanan data sementara yang bertipe sama di dalam memori, sebelum data tersebut diproses lebih lanjut.
 - Menghemat penggunaan nama variable yang banyak (*yang bertipe sama*).
→ karena sebuah variable hanya dapat menampung sebuah nilai
 - Mengurangi penulisan perintah yang sama secara berulang ulang untuk variable-variable tersebut.

DENGAN Array	TANPA Array
<pre>for(i=1;i<=nEf;i++) { printf("%c", nilai[i]); }</pre>	<pre>printf("%c", nilai1); printf("%c", nilai2); printf("%c", nilai3); printf("%c", nilai4);</pre>



Bekerja dengan Dua Buah Array

- Dua buah array atau lebih dapat dioperasikan sekaligus.

Contoh :

Nilai ujian N mahasiswa disimpan dalam array UJIAN[1..N]. Nilai tersebut akan dikonversi menjadi nilai indeks (A/B/C/D/E) dan disimpan pada array INDEKS[1..N].

- Deklarasi Data :

- ```
#define N 70
typedef int Ujian[N];
typedef char Indeks[N];
```





# Bekerja dengan Dua Buah Array

- Prosedur untuk menghitung nilai indeks:

```
void NilaiToIndeks(Ujian nilai, int nEf, Indeks *indeks)
{
 int i;

 for(i=1;i<=nEf;i++)
 {
 if(nilai[i]>=80)
 (*indeks)[i]='A';
 else if(nilai[i]>=60)
 (*indeks)[i]='B';
 else if(nilai[i]>=40)
 (*indeks)[i]='C';
 else if(nilai[i]>=20)
 (*indeks)[i]='D';
 else
 (*indeks)[i]='E';
 }
}
```



# Question and Answer



# Terima kasih



uajy



Universitas Atma Jaya Yogyakarta



[www.uajy.ac.id](http://www.uajy.ac.id)