

Algoritme dan Pemrograman

Kuliah #8
Array

Array

- Dalam matematika dikenal variabel berindeks $x_0, x_1, x_2, \dots, x_{n-1}$. Angka 0, 1, 2, ..., n-1 pada variabel x disebut sebagai indeks atau subscript.
- Variabel berindeks diimplementasikan dalam program berupa *array* $x[0], x[1], x[2], \dots, x[n-1]$, dengan n adalah ukuran *array*.
- Selang nilai yang valid bagi indeks suatu variabel array berukuran n adalah 0 sampai dengan n-1.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array

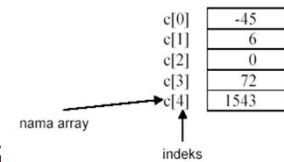
- Definisi: kumpulan elemen bertipe data sama yang diakses dengan nama yang sama
- Deklarasi:
`data-type array-name[size];`
- Array dapat dibedakan menjadi:
 - Array berdimensi satu (1D) \cong vektor di Matematika
 - Array berdimensi dua (2D) \cong matriks di Matematika
 - Array berdimensi banyak
- Elemen *array* disimpan di memori secara berurutan

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array satu dimensi (1D)

- Contoh:

```
int a[5];
float x[10];
int d[3]={0};
int c[]={-45, 6, 0, 72, 1543};
```

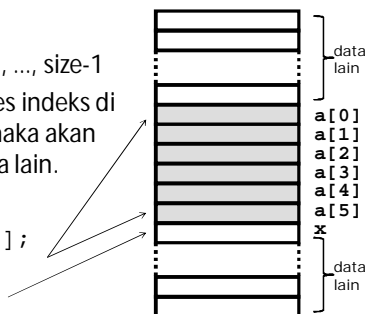


DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Alokasi memori

- Indeks: 0, 1, 2, ..., size-1
- Jika mengakses indeks di luar selang, maka akan diperoleh data lain.
- Contoh:

```
int a[6];
int x;
```



DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Alokasi memori

- *Array* harus dideklarasikan terlebih dahulu.
- *Array* dapat dideklarasikan secara global di luar fungsi main, maupun secara lokal di dalam suatu fungsi.

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Mengisi masukan ke *array*

```
#include <stdio.h>
int main() {
    int a[100];
    int i;

    scanf("%d", &n); // baca banyaknya data

    // membaca masukan dan disimpan ke array
    for (i=0; i<n; i++)
        scanf("%d", &a[i]);

    printf("%d\n", a[n/2]); // akses isi array
    return 0;
}
```

Contoh masukan: 3 20 34 21

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Keluaran?

```
#include <stdio.h>

int main() {
    int a[3], b[4] = {4,5,2,3};
    int c[] = {3,2,1}, d[4] = {0};
    int e[5] = {1}, f[];

    printf("%d %d %d\n", a[0], b[1], c[2]);
    printf("%d %d %d\n", d[3], e[4], f[5]);
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Penjelasan

- Gagal kompilasi karena dimensi *f* tidak dinyatakan baik secara eksplisit maupun implisit
 - Berikan dimensi, misalnya: `f[10]`
- Setelah diperbaiki:
 - `a[0]` tidak diketahui karena tidak diinisialisasi
 - `b[1] = 5`, `c[2] = 1`, `d[4] = 0`, `e[5] = 0`
- Jika hanya sebagian yang diinisialisasi, sisanya bernilai 0

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Menghitung jumlah isi *array*

```
#include <stdio.h>
#define SIZE 5

int main() {
    int a[SIZE] = {1, 3, 5, 4, 7};
    int i, total = 0;

    for (i=0; i<SIZE; i++)
        total += a[i];

    printf("%d\n", total );

    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Akses *array* di luar batas indeks

```
#include <stdio.h>
int main() {
    int a[5] = {1, 3, 5, 4, 7};
    printf("%d\n", a[6] );
    return 0;
}
```

D:\My Documents\2088763392

Apakah terjadi *error* pada saat di-*compile* atau di-*run*? Mengapa hal tersebut terjadi?

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan

1. Program untuk menghitung rata-rata nilai dalam suatu *array*
2. Program untuk menentukan keberadaan suatu nilai dalam suatu *array*
3. Program untuk menentukan nilai minimum dan maksimum dalam suatu *array*

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array sebagai argumen fungsi

- Seperti variabel lainnya, *array* juga dapat digunakan sebagai argumen dari fungsi.
- Nilai variabel *array* yang dideklarasikan sebagai parameter fungsi akan mempengaruhi nilai variabel *array* pada fungsi pemanggilnya (*call by reference*) (dibahas dalam topik *pointer*).
- Contoh deklarasi *array* dalam fungsi:


```
int printarray(int n, int a[n]) {
    ...
}
```
- Perhatikan bahwa parameter *n* harus disebut sebelum *a[n]*

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi Mencetak Elemen Array

```
#include <stdio.h>
void printarray(int n, int a[n]);
// atau: void printarray(int n, int a[]);

int main() {
    int x[5] = {1,4,2,-3,9};
    printarray(x,5);
    return 0;
}

void printarray(int n, int a[n], int n) {
    int i;
    for (i=0; i<n; i++)
        printf("%d ", a[i]);
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pengiriman Array 1D ke Fungsi

- Dimensi *array* 1D pada *function header* tidak harus dinyatakan


```
void printarray(int a[], int n)
atau
void printarray(int a*, int n)
```
- Jika dimensi tidak dinyatakan, tidak harus muncul setelah parameter penampung dimensi (*n* pada contoh)
- Ukuran *array* menjadi parameter (*n* pada contoh)
- Tipe **void** digunakan jika fungsi tidak mengembalikan suatu nilai
 - Serupa dengan prosedur dalam terminologi bahasa pemrograman lain

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi Menjumlahkan Elemen Array

```
#include <stdio.h>
int sumarray(int a[], int a);

int main() {
    int x[5] = {1,4,2,-3,9};
    printf("%d", sumarray(x,5));
    return 0;
}

int sumarray(int a[], int n) {
    int i, sum = 0;
    for (i=0; i<n; i++) sum+=a[i];
    return sum;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi Membalik Elemen Array

```
void fliparray(int a[], int n) {
    int i, temp;
    for (i=0; i<n/2; i++){
        temp = a[i];
        a[i] = a[n-i-1];
        a[n-i-1] = temp;
    }
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Keluaran?

```
int main() {
    int x[] = {5,2,-1,7,-3};
    printarray(x, 5);
    fliparray(x,5);
    printarray(x, 5);
    fliparray(x,3);
    printarray(x, 5);
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array dua dimensi (2D)

- Dapat disepadankan dengan matriks
- *Array* yang terdiri atas *array*
- Contoh:

```
int b[][] = {{2,4,1}, {5,3,7}};
```

	0	1	2
0	2	4	1
1	5	3	7

```
float x[3][4];
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array 2D: Membaca matriks m×n

```
int a[100][100];
int i, j;

// minta dimensi masukan
scanf("%d %d", &m, &n);

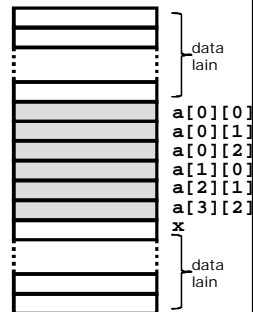
// membaca elemen matriks
for (i=0; i<m; i++)
    for (j=0; j<n; j++)
        scanf("%d", &a[i][j]);
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Array 2D di memori

- Elemen berurutan di memori
- Misal:

```
int a[2][3]
```
- Seperti *array* 1D berapa elemen?



DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Pengiriman Array 2D ke Fungsi

- Dimensi kedua untuk *array* 2D pada *function header* harus dinyatakan: `a[m][n]`, `a[*][*]`, `a[][*]`, atau `a[][n]`
- Ukuran *array* menjadi parameter (`m` dan `n` pada contoh)
- *Array* 2D dapat dikirim ke fungsi yang mengharapkan *array* 1D
 - Ingat *array* 2D = *array* yang elemennya adalah *array*
 - Elemen 2D juga berurutan

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Keluaran?

```
#include <stdio.h>

int main() {
    int x[5][4], i, j, k=1;

    for(i=0; i<5; i++)
        for(j=0; j<4; j++) x[i][j]=k++;

    printarray(x, 20);
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Perhatikan fungsi berikut yang dimaksudkan mencetak matrix

```
#define M 100
#define N 100

void printmatrix0
(int a[M][N], int m, int n) {
    int i, j;
    for (i=0; i<m; i++)
        for (j=0; j<n; j++) {
            printf("%d ", a[i][j]);
            printf((n-1)==j ? "\n" : " ");
        }
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Keluaran?

```
#include <stdio.h>
#define M 100
#define N 100

int main() {
    int x[5][4], i, j, k=1;

    for(i=0; i<5; i++)
        for(j=0; j<4; j++) x[i][j]=k++;

    printmatrix0(x, 5, 4);
    return 0;
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Bandingkan dengan

```
void printmatrix1
(int m, int n, int a[m][n]) {
    int i, j;
    for (i=0; i<m; i++)
        for (j=0; j<n; j++) {
            printf("%d", a[i][j]);
            printf((n-1)==j ? "\n" : " ");
        }
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

atau

```
#define M 100
#define N 100

void printmatrix2
(int a[M][N], int m, int n) {
    int i=0, j=0, x = m*n;
    while (x--) {
        printf("%d", a[i][j]);
        j++;
        if(j==N) (j=0, i++);
        printf(0==(j%n)? "\n" : " ");
    }
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Penjelasan

- Fungsi `printmatrix0` dan `printmatrix2` mengharapkan parameter berupa *array* 100x100 (karena M dan N diganti dengan 100)
- Bandingkan `a[100][100]` dan `b[5][4]`:
 - Urutan `a[0][0]` s/d `a[0][3]` sama dengan urutan `b[0][0]` s/d `b[0][3]`, tetapi
 - Elemen berikutnya di `a` adalah `a[0][4]` sedangkan di `b` adalah `b[1][0]`
- Hasilnya, keluaran `printmatrix0` pada hanya benar untuk baris pertama
- Pada `printmatrix2` indeksnya disesuaikan dengan *array* 100x100

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Catatan

- Fungsi `printmatrix1` dan `printmatrix2` hanya benar untuk mencetak semua elemen *array* 2D
- Misal `a[2][4]`:


```
1 2 3 4
5 6 7 8
```

Mencetak dengan `m = 2` dan `n = 3`:

```
1 2 3
4 5 6
```
- Bukan:


```
1 2 3
5 6 7
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Fungsi memutar matriks

```
void putar
(int m, int n, int a[m][n], int b[n][m])
{
    int i, j;
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
            b[j][i]=a[i][j];
}
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

