



Bahasa C Array

Slide diambil dari MK Dasar-Dasar
Pemrograman Fasilkom UI

Dimodifikasi dan disesuaikan dngn
kebutuhan pengajaran di PTI-Undiksha



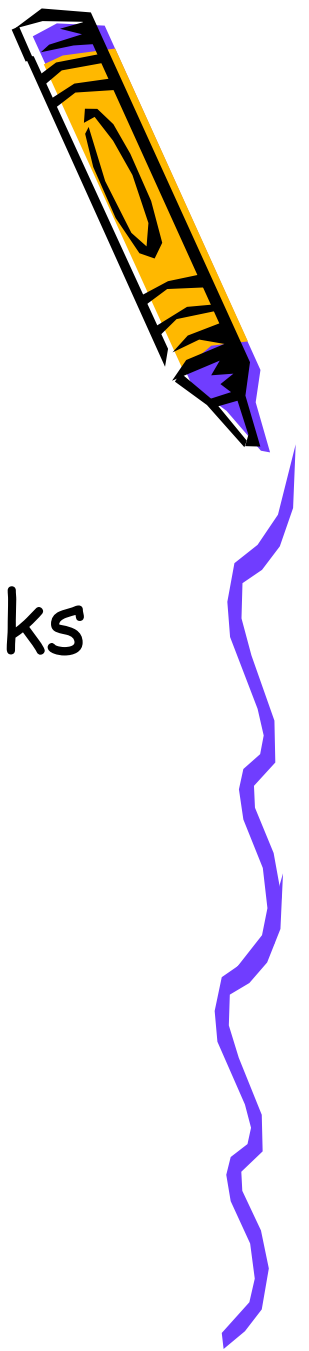
Pengenalan Array



- ❑ Array (larik) merupakan suatu struktur data yang terdiri dari beberapa data bertipe sama.
- ❑ Suatu array menempati sekelompok lokasi memori yang berurutan. Tiap lokasi ini berlaku sebagai suatu variabel dengan tipe yang sama.



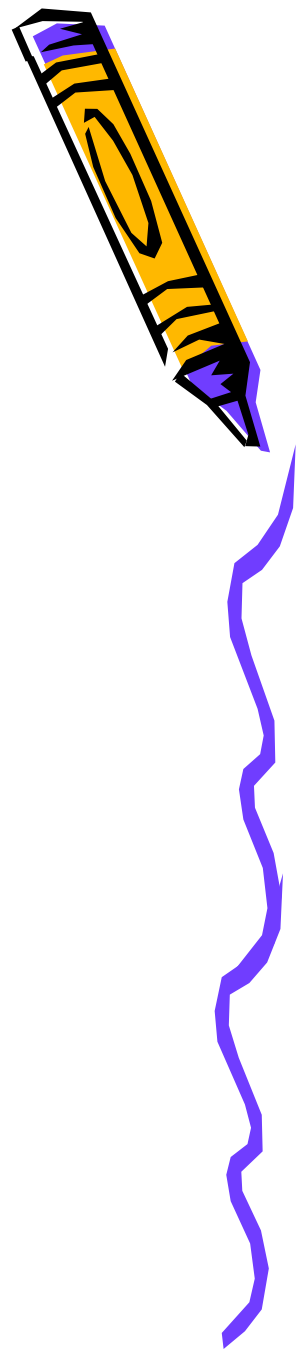
Pengenalan Array



- ☐ Unsur dari suatu array dapat diakses secara langsung dengan menggunakan nama array dan indeks (subskrip) yang bersangkutan
- ☐ Penamaan array mengikuti aturan penamaan variabel



Name of array (Note that
all elements of this
array have the same name,
c)



□ Deklarasi
array bernama
c yang terdiri
dari 12 unsur
bertipe
integer:

int c[12]


↓	c[0]	-45
	c[1]	6
	c[2]	0
	c[3]	72
	c[4]	1543
	c[5]	-89
	c[6]	0
	c[7]	62
	c[8]	-3
	c[9]	1
	c[10]	6453
	c[11]	78

↑
Position number of the
element within array c




Contoh: Inisialisasi Array

```
#include <stdio.h>
int main()
{
    int n[ 10 ]; /* n is an array of 10 integers */
    int i;
    for ( i = 0; i < 10; i++ )
        n[ i ] = 0; /* set element at location i to 0 */
    printf( "%s%13s\n", "Indeks", "Nilai" );
    for ( i = 0; i < 10; i++ )
        printf( "%7d%13d\n", i, n[ i ] );
    return 0;
}
```



Indeks	Nilai
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0



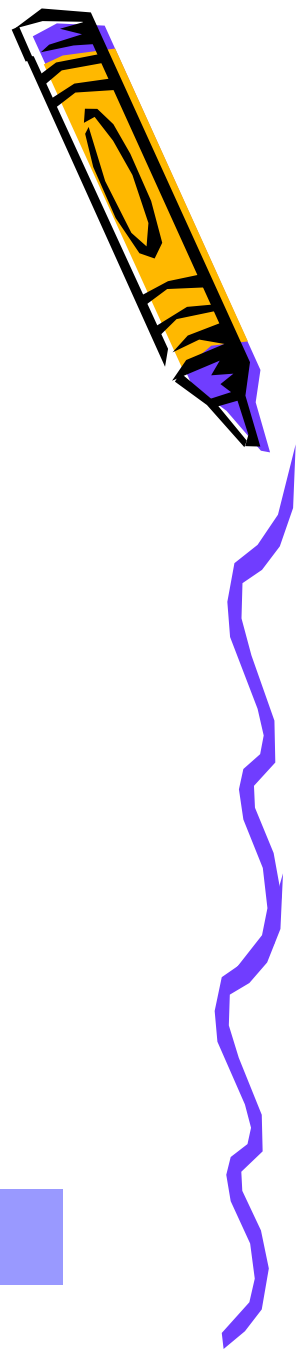
Contoh: Inisialisasi Array

```
#include <stdio.h>
int main()
{
    int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
    int i;
    printf( "%s%13s\n", "Indeks", "Nilai" );
    for ( i = 0; i < 10; i++ )
        printf( "%7d%13d\n", i, n[ i ] );
    return 0;
}
```

Indeks	Nilai
0	32
1	27
2	64
3	18
4	95
5	14
6	90
7	70
8	60
9	37



Contoh: Menghitung jumlah unsur- unsur array



```
#include <stdio.h>
#define UKURAN 10
int main(){
    int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
    int i, jumlah = 0;

    for ( i = 0; i <= UKURAN-1; i++ )
        jumlah += n[i];

    printf( "Jumlah nilai elemen array %d", jumlah);
    return 0;
}
```



Jumlah nilai elemen array 507

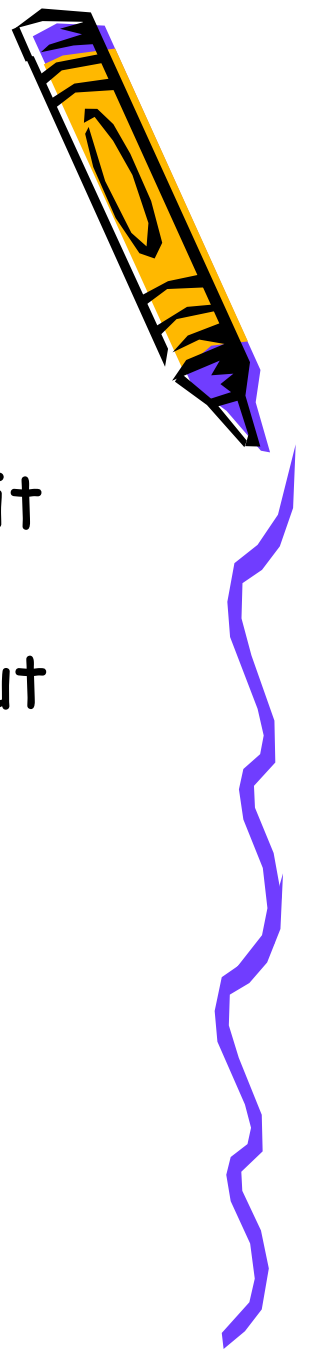
Inisialisasi Array



- ❑ Array dapat diinisialisasi secara eksplisit pada waktu deklarasi. Nilai-nilai awal diapit oleh kurung kurawal { }. Apabila banyaknya nilai awal kurang dari ukuran array, maka unsur-unsur array yang tersisa diberi nilai awal nol.
- ❑ Contoh:
 - `int a[50] = {0};`
→ semua unsur dari array a diberi nilai 0
 - `int b[10] = {-1, 3};`
→ unsur `b[0]` diberi nilai -1, `b[1]` diberi nilai 3, dan selain itu diberi nilai 0



Inisialisasi Array



- ❑ Jika suatu array dideklarasikan tanpa ukuran dan diinisialisasi secara eksplisit dengan sebarisan nilai, maka compiler akan menentukan ukuran array tersebut sama dengan banyaknya nilai awal.
- ❑ Contoh:
 - `int a[] = {2, 2, 3, 3}` EKUIVALEN dengan
`int a[4] = {2, 2, 3, 3}`



Array of Char

- ❑ String (kata), seperti "abc" adalah suatu array yang terdiri dari karakter-karakter yang diakhiri karakter NULL ('\0').

char s[] = "abc"; EKUIVALEN dengan

char s[] = {'a', 'b', 'c', '\0'}; EKUIVALEN dengan

char s[4] = {'a', 'b', 'c', '\0'};

- ❑ Jadi, string "abc" terdiri dari 4 karakter, Nilai dari sizeof("abc") adalah 4



Array of Char



- ❑ 'x' tidak sama dengan "x". Apa bedanya?
- ❑ Perintah `scanf` dapat dipakai untuk menginput suatu string dari keyboard dengan penentu konversi `%s`. String itu disimpan dalam suatu array char.
- ❑ Suatu string char yang merepresentasikan suatu string dapat dioutput dengan `printf` melalui penentu konversi `%s`



Contoh penggunaan string & array char

```
#include <stdio.h>
main() {
    char str1[20], str[] = "Ilmu Komputer";
    int i;
    printf "Masukkan sebuah string:\n";
    scanf ("%s", str1);
    printf ("String1: %s\nString2: %s\n", str1, str);

    printf ("String1 dengan spasi antar karakter:\n");
    for (i=0; str1[i] != '\0'; i++)
        printf ("%c ", str1[i]);
    return 0;
}
```

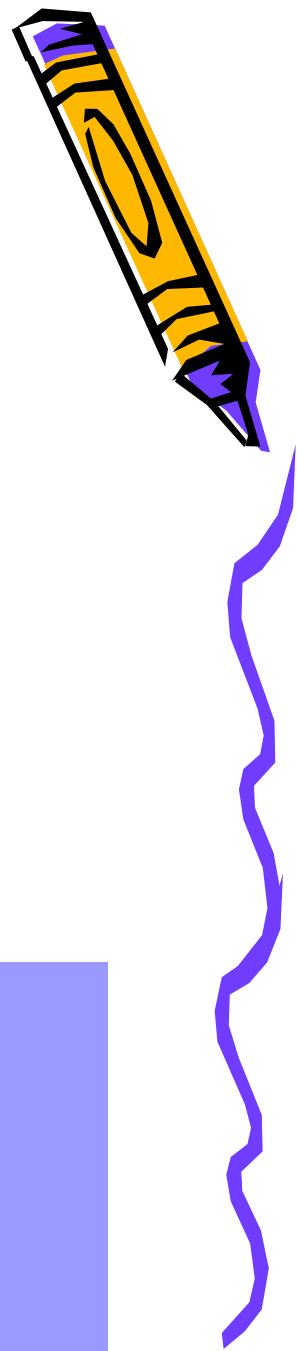
Masukkan sebuah string: Fasilkom

String1: Fasilkom

String2: Ilmu Komputer

String1 dengan spasi antar karakter:

F a s i l k o m



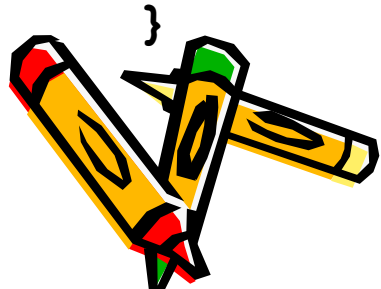
Contoh Program: Student Poll Program



```
#include <stdio.h>
#define RESPONSE_SIZE 40 /* define array sizes */
#define FREQUENCY_SIZE 11
int main() {
    int answer; /* counter to loop through 40 responses */
    int rating; /* counter to loop through frequencies 1-10 */
    int frequency[ FREQUENCY_SIZE ] = { 0 };
    int responses[ RESPONSE_SIZE ] = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
                                         1, 6, 3, 8, 6, 10, 3, 8, 2, 7, 6, 5, 7, 6, 8,
                                         6, 7, 5, 6, 6, 5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };
    for ( answer = 0; answer < RESPONSE_SIZE; answer++ )
        ++frequency[ responses [ answer ] ];

    printf( "%s%17s\n", "Rating", "Frequency" );

    for ( rating = 1; rating < FREQUENCY_SIZE; rating++ )
        printf( "%6d%17d\n", rating, frequency[ rating ] );
    return 0;
}
```



Contoh Program: Student Poll Program



Output:

Rating	Frequency
1	2
2	2
3	2
4	2
5	5
6	11
7	5
8	7
9	1
10	3



Contoh Program: Histogram Printing



```
#include <stdio.h>
#define SIZE 10

int main() {
    int n[ SIZE ] = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
    int i; /* outer for counter for array elements */
    int j; /* inner for counter counts *s in each histogram bar */

    printf( "%s%13s%17s\n", "Element", "Value", "Histogram" );

    for ( i = 0; i < SIZE; i++ ) {
        printf( "%7d%13d", i, n[ i ] );

        for ( j = 1; j <= n[ i ]; j++ ) {
            printf( "%c", '*' );
        }

        printf( "\n" );
    }

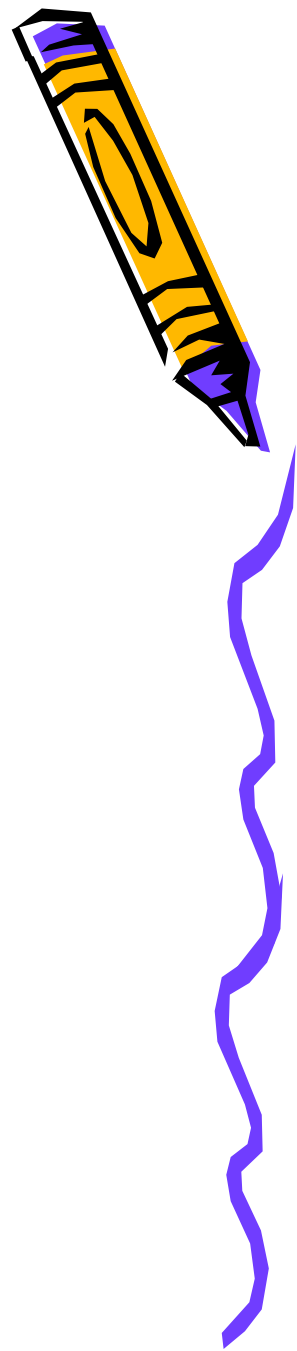
    return 0;
}
```



Contoh Program: Histogram Printing

Output:

Element	Value	Histogram
0	19	*****
1	3	***
2	15	*****
3	7	*****
4	11	*****
5	9	*****
6	13	*****
7	5	*****
8	17	*****
9	1	*



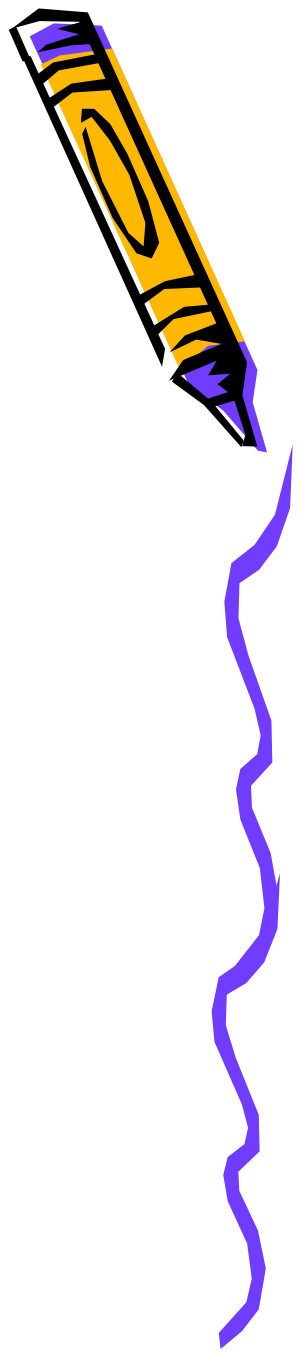
Contoh Prog: Melempar Dadu 6000 x

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 7

int main() {
    int face; /*nilai random dadu 1 - 6 */
    int roll; /* counter pelemparan*/
    int frequency[ SIZE ] = { 0 }; /* clear counts */

    srand( time( NULL ) ); /* seed random-number generator */

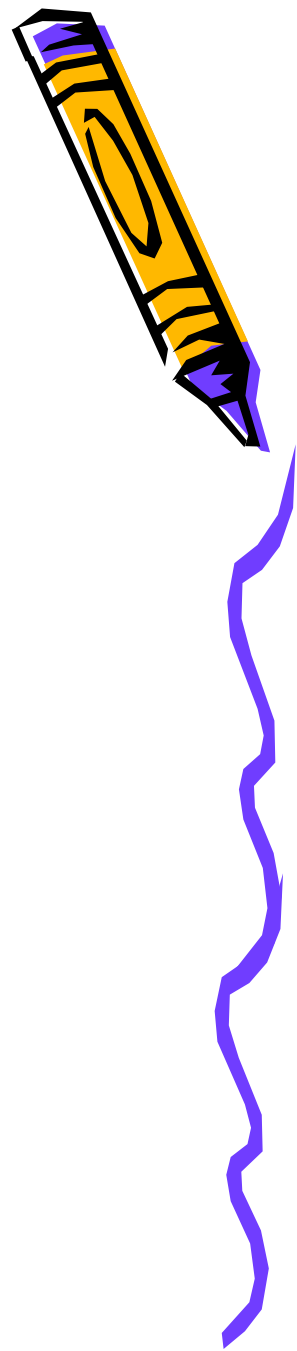
    for ( roll = 1; roll <= 6000; roll++ ) {
        face = 1 + rand() % 6;
        ++frequency[ face ];
    }
    printf( "%s%17s\n", "Face", "Frequency" );
    for ( face = 1; face < SIZE; face++ )
        printf( "%4d%17d\n", face, frequency[ face ] );
    return 0;
}
```



Contoh Prog: Melempar Dadu 6000 x

Output

Face	Frequency
1	1029
2	951
3	987
4	1033
5	1010
6	990



Array sebagai argumen fungsi



- ❑ Prototipe fungsi:

`int maksimum (int a[], int ukuran)`

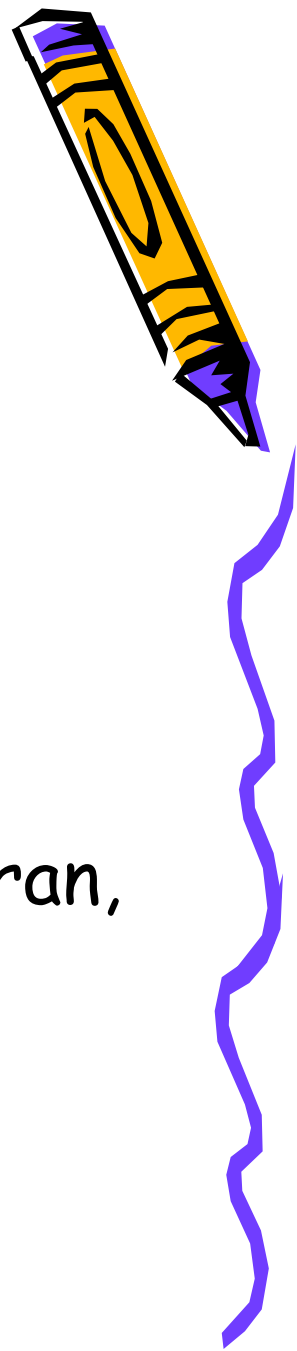
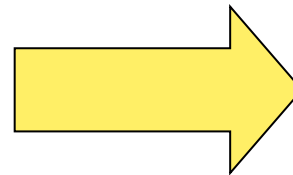
menyatakan bahwa fungsi maksimum memerlukan dua argumen. Argumen pertama suatu array bertipe int, argumen kedua suatu int yang merupakan ukuran array tersebut.

- ❑ Sebagai argumen aktual, nama array saja yang dipakai.

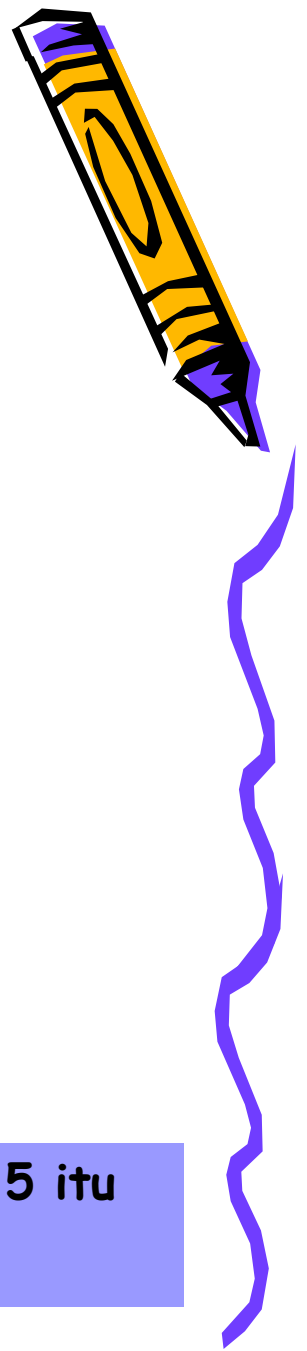


Contoh penggunaan array sbg argumen

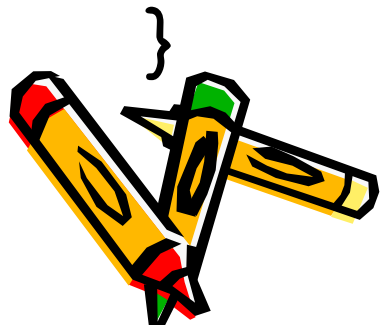
```
#include <stdio.h>
int maksimum (int a[], int uk);
main(){
    int ukuran, a[] = {9,0,3,4,5};
    ukuran = sizeof(a) / sizeof(int);
    printf ("Nilai maksimum dalam array
    berukuran %d itu \nadalah %d\n", ukuran,
    maksimum(a, ukuran));
    return 0
```



Contoh penggunaan array sbg argumen



```
/* Fungsi untuk mengembalikan nilai  
maksimum dalam array */  
int maksimum (int a[], int uk) {  
    int i, maks;  
    maks = a[0];  
    for (i=1; i < uk; i++)  
        if (a[i] > maks) maks = a[i];  
    return maks;  
}
```



Nilai maksimum dalam array berukuran 5 itu
adalah 9