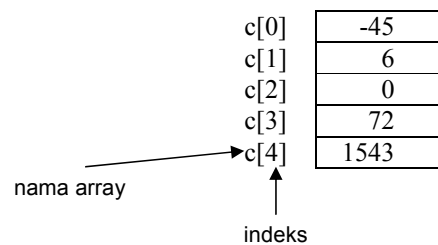


BAB VI ARRAY

Pada program yang dibahas terdahulu banyak menggunakan variabel tunggal, artinya sebuah variabel hanya digunakan untuk menyimpan satu nilai. Array adalah sekelompok lokasi memori dimana semuanya mempunyai nama dan tipe variabel yang sama. Berikut adalah contoh variabel bernama `c` yang mempunyai 5 lokasi memori yang semuanya bertipe `int`.



Masing-masing nilai dalam setiap lokasi mempunyai identitas berupa nama ('c') dan nomor indeks yang dituliskan di dalam tanda kurung siku '[' ...]'. Sebagai contoh, 72 adalah nilai dari `c[3]`. Nilai indeks ini merupakan suatu ekspresi, sehingga jika `a=1` dan `b=2`, maka `c[a+b] += 2` berarti menambahkan nilai 2 ke dalam `c[3]`.

DEKLARASI ARRAY

Variabel array dideklarasikan dengan mencantumkan tipe dan nama variabel yang diikuti dengan banyaknya lokasi memori yang ingin dibuat. Dengan demikian, deklarasi untuk variabel array `c` di atas adalah:

```
int c[5];
```

Perlu diperhatikan bahwa C secara otomatis akan menyediakan lokasi memori sesuai dengan yang dideklarasikan, dimana nomor indeks selalu dimulai dari 0.

Nilai suatu variabel array dapat juga diinisialisasi secara langsung pada saat deklarasi, misalnya:

```
int c[5] = {-45, 0, 6, 72, 1543};
```

berarti setiap lokasi memori dari variabel array `c` langsung diisi dengan nilai-nilai yang dituliskan di dalam tanda kurung kurawal. Contoh berikut adalah deklarasi variabel array sekaligus mengisi setiap lokasi memorinya dengan nilai 0:

```
int x[5] = {0};
```

Banyaknya lokasi memori dapat secara otomatis disediakan sesuai dengan banyaknya nilai yang akan dimasukkan, seperti contoh berikut yang tentunya membuat variabel array dengan 10 lokasi memori:

```
int x[] = {10, 15, 12, 5, 13, 9, 6, 17, 25, 31};
```

Berikut adalah contoh aplikasi variabel array, yaitu program untuk menghitung jumlah setiap elemen dalam suatu array.

```
#include <stdio.h>
#define SIZE 12

main() {
    int a[SIZE] = {1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45};
    int indeks, total = 0;

    for (indeks=0 ; indeks<=SIZE-1 ; indeks++)
        total += a[indeks];

    printf("\nTotal setiap elemen array adalah %d", total);
    return 0;
}
```

Variabel array juga dapat dipakai untuk menyederhanakan perhitungan yang berstruktur, misalnya menghitung frekuensi bilangan (1 hingga 10) yang muncul tanpa menggunakan struktur IF seperti pada program berikut.

```
#include <stdio.h>
#define dSIZE 40
#define fSIZE 11

main() {
    int indeks, jawaban;
    int respon[dSIZE] = {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};
    int freq[fSIZE] = {0};

    for (indeks=0 ; indeks<=dSIZE-1 ; indeks++)
        ++freq[respon[indeks]];

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

    for (jawaban=1 ; jawaban<=fSIZE-1 ; jawaban++)
        printf("%6d%17d\n", jawaban, freq[jawaban]);
    return 0;
}
```

MENGIRIM ARRAY KE FUNGSI

Mengirimkan argumen array ke sebuah fungsi dilakukan dengan hanya mencantumkan nama array tanpa diikuti dengan tanda apapun, seperti contoh berikut:

```
int c[5] = {-45, 0, 6, 72, 1543};
.....
.....
JUMLAH(c, 5);
.....
```

yang memanggil fungsi JUMLAH dengan mengirimkan argumen berupa variabel array c dan sebuah konstanta 5. Perhatikan bahwa variabel array ditulis hanya c, tanpa notasi tambahan apapun. Deklarasi variabel array yang menjadi parameter dari suatu fungsi dituliskan dengan nama variabel array yang diikuti dengan tanda kurung [], tanpa menuliskan banyaknya lokasi memori yang diinginkan.

Program berikut memberi contoh penggunaan variabel array dalam suatu fungsi dan cara pemanggilannya.

```
#include <stdio.h>
#define SIZE 99

void SORT(int [], int);

main() {
    int x[SIZE], counter, n;

    printf("\nKetiklah banyaknya data :"); scanf("%d", &n);    /* baca n */
    for (counter=1 ; counter<=n ; counter++) {
        printf("\nKetik data ke-%d : ",counter);
        scanf("%d", &x[counter-1]);    /* baca array */
    }

    SORT(x, n);    /* panggil fungsi */

    printf("\nData setelah SORTING");
    for (counter=1 ; counter<=n ; counter++)
        printf("\n%d - %d", counter, x[counter-1]);    /* mencetak hasil */

    return 0;
}

void SORT(int a[], int size) {    /* fungsi sorting */
    int pass, j, hold;
    for (pass=1 ; pass<=size-1 ; pass++)
        for (j=0 ; j<=size-2 ; j++)
            if (a[j] > a[j+1]) {
                hold = a[j]; a[j] = a[j+1]; a[j+1] = hold;
            }
}
```

ARRAY GANDA

Struktur array yang dibahas terdahulu mempunyai satu dimensi, sehingga variabelnya disebut dengan variabel array berdimensi satu. Pada bagian ini ditunjukkan array

berdimensi lebih dari satu, yang sering disebut dengan array berdimensi ganda. Sebagai contoh, sebuah matrik B berukuran 2 x 3 dapat dideklarasikan dalam C seperti berikut:

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

yang akan menempati lokasi memori dengan susunan sebagai berikut:

	0	1	2
0	2	4	1
1	5	3	7

dan definisi variabel untuk setiap elemen tersebut adalah:

	0	1	2
0	b[0][0]	b[0][1]	b[0][2]
1	b[1][0]	b[1][1]	b[1][2]

Sebagai contoh, berikut adalah program untuk lebih memahami konsep array berganda, dan cobalah menentukan keluaran dari program tersebut.

```
#include <stdio.h>

void printArray(int [][][3]);

main() {
    int matrik1[2][3] = { {1, 2, 3}, {4, 5, 6} },
        matrik2[2][3] = { 1, 2, 3, 4, 5 },
        matrik3[2][3] = { {1, 2}, {4} };
    printArray(matrik1);
    printArray(matrik2);
    printArray(matrik3);
    return 0;
}

void printArray(int a[][][3]) {
    int i, j;

    for (i=0 ; i<=1 ; i++) {
        for (j=0 ; j<=2 ; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
}
```

STRING

Suatu string sepanjang n adalah array yang terdiri dari n karakter penyusunnya dan diakhiri dengan sebuah *null character* yang dilambangkan dengan '\0'. Dengan demikian, string "komputer" adalah terdiri dari array dengan isi sebagai berikut:

```
'k' 'o' 'm' 'p' 'u' 't' 'e' 'r' '\0'
```

Sehingga deklarasi variabel string dapat dilakukan dengan cara-cara sebagai berikut:

```
char s1[]="komputer";  
char s2[]={ 'k', 'o', 'm', 'p', 'u', 't', 'e', 'r', '\0'};
```

LATIHAN 6.

1. Tuliskan nilai setiap elemen dari variabel array a di bawah ini

```
int a[3][3] = { {1, 8}, {2, 4, 6}, {5} };
```

2. Buatlah sebuah fungsi bernama testPalindrome untuk menguji apakah sebuah string bersifat palindrome, artinya dibaca dari kiri sama dengan dibaca dari kanan. Fungsi akan menghasilkan nilai 1 jika benar, dan 0 bila tidak.
3. Buatlah fungsi rekursif untuk mendapatkan bilangan terkecil dari n buah bilangan bulat yang tersimpan dalam sebuah variabel array.
4. Buatlah fungsi rekursif untuk membalik isi sebuah variabel array.
5. Apa yang dilakukan oleh program berikut:

```
#include <stdio.h>  
#define SIZE 10  
  
int whatIsThis(int [], int);  
  
main() {  
    int total, a[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
    total = whatIsThis(a, SIZE);  
    printf("\nNilai variabel total adalah %d", total);  
    return 0;  
}  
  
int whatIsThis(int b[], int size) {  
    if (size == 1)  
        return b[0];  
    else  
        return b[size-1] + whatIsThis(b, size-1);  
}
```