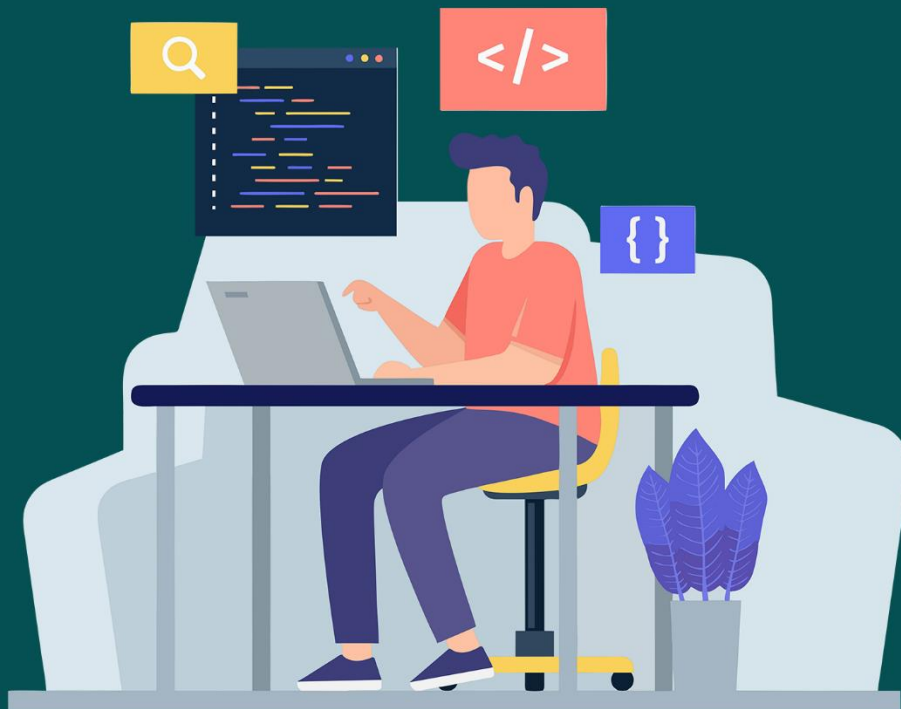


Modul Praktikum  
Algoritma dan Pemrograman I

## **“Array dan Implementasi For dalam Array”**



TIM ASISTEN PEMROGRAMAN  
ANGKATAN 12

Departemen Pendidikan Ilmu Komputer  
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam  
Universitas Pendidikan Indonesia  
2021

## Pendahuluan

**Array** adalah kumpulan dari nilai-nilai data yang bertipe sama. Jika digambarkan dalam logika sehari-hari, maka array adalah sebuah kereta yang tersusun dari beberapa gerbong, dengan kereta melambangkan variabelnya dan gerbong sebagai indeksinya.

Array dideklarasikan seperti variabel biasa, tetapi diakhiri dengan alokasi array dalam kurung siku.

Contoh : `char arr[6];`

Dalam kebanyakan bahasa pemrograman, indeks atau nomor urutan array dimulai dari 0 dan berurut hingga (n-1), dengan n adalah panjang atau alokasi yang diberikan.

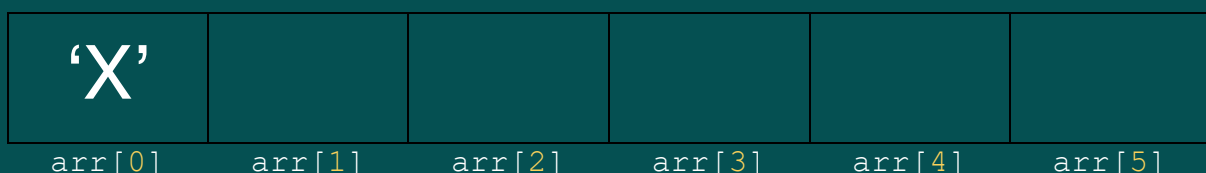


Ilustrasi tersebut menunjukkan array dengan alokasi sebanyak 6. Meskipun penulisannya indeksinya hanya sampai 5, namun jumlahnya tetap ada 6 karena indeksinya dimulai dari 0.

Pengisian nilai dari array sama seperti pengisian nilai variabel biasa, hanya saja perlu ditunjuk indeks mana yang akan diisi. Misalkan program harus memasukkan nilai X pada indeks ke-0, maka penulisan kodenya adalah:

```
arr[0] = 'X';
```

Jika program dijalankan, maka ilustrasi array akan berubah menjadi:



Selain itu, nilai dari array juga dapat diisi sekaligus hingga indeks terakhir, selama masih berada dalam baris kode deklarasinya. Misal:

```
char arr[5] = {'T', 'e', 'c', 'h', 'i'};
```

Maka ilustrasi array akan menjadi:

'T'	'e'	'c'	'h'	'i'
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

Contoh program:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      // Deklarasi array secara langsung.
6      char name[5] = {'T', 'E', 'C', 'H', 'I'};
7
8      // Deklarasi array secara bertahap.
9      int num[3];
10     num[0] = 38;
11     num[1] = 46;
12     num[2] = 48;
13
14     // Keluaran.
15     printf("Name : \n");
16     printf("%c %c %c %c %c\n", name[0], name[1], name[2], name[3], name[4]);
17     printf("Num  : \n");
18     printf("%d %d %d\n", num[0], num[1], num[2]);
19
20     return 0;
21 }
```

Keluaran:

```
Name :
T E C H I
Num  :
38 46 48
```

**\* Pastikan banyaknya data yang dimasukkan sama dengan alokasi array.**

## Penggunaan Array Dinamis

Sejauh ini, penulisan alokasi array selalu ditentukan secara statis (fixed). Tentunya hal ini dianggap kurang efisien, terutama jika penggunaan memori array tidak sebanyak alokasi array.

Sebagai contoh, program sudah mengalokasikan array sebanyak 100, tetapi yang digunakan hanya 5. Maka sisa alokasi yang sebanyak 95 itu akan sia-sia.

Oleh karena itu, digunakanlah bentuk array dinamis. **Array dinamis** adalah **array yang alokasinya menyesuaikan jumlah yang diinginkan**. Hal ini dapat meminimalisir kasus alokasi yang sia-sia tadi, karena jumlah indeks diatur sesuai kebutuhan.

Contoh program:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      // Deklarasi masukan untuk alokasi array.
6      int n;
7      scanf("%d", &n);
8
9      // Deklarasi array sesuai alokasi yang telah ditentukan.
10     int arr[n];
11
12     return 0;
13 }
```

**\* Pastikan nilai n dimasukkan terlebih dahulu sebelum mendeklarasikan array.** Dengan begitu, alokasi array akan menyesuaikan nilai n yang dimasukkan.

## Penggunaan Array Dinamis

Masalah alokasi array selesai. Akan tetapi, bagaimana cara memasukkan nilai ke setiap indeks-nya? Jika kurang dari alokasi, maka tak ada bedanya dengan array statis. Jika lebih dari alokasi, akan terjadi **segmentation fault** (error karena mengakses memori yang seharusnya tidak boleh diakses).

Di sinilah, **implementasi perulangan for digunakan**. Dengan inisialisasi 0 dan kondisi perulangan lebih kecil dari alokasi array, maka setiap indeks dalam array akan **pasti** terisi.

Contoh program, jika ingin mengisi nilai array dari 1 hingga n:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      // Deklarasi variabel kontrol dan masukan untuk alokasi array.
6      int i, n;
7
8      printf("Masukkan alokasi array : ");
9      scanf("%d", &n);
10
11     // Deklarasi array sesuai alokasi yang telah ditentukan.
12     int arr[n];
13
14     // Perulangan untuk memasukkan nilai array.
15     for(i = 0; i < n; i++)
16     {
17         arr[i] = (i + 1);
18     }
19
20     // Keluaran.
21     for(i = 0; i < n; i++)
22     {
23         printf("Nilai array indeks ke-%d : %d\n", i, arr[i]);
24     }
25
26     return 0;
27 }
```

Keluaran:

```
Masukkan alokasi array : 6
Nilai array indeks ke-0 : 1
Nilai array indeks ke-1 : 2
Nilai array indeks ke-2 : 3
Nilai array indeks ke-3 : 4
Nilai array indeks ke-4 : 5
Nilai array indeks ke-5 : 6
```

Atau, jika ingin nilai setiap indeksnya juga ditentukan pengguna:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      // Deklarasi variabel kontrol dan masukan untuk alokasi array.
6      int i, n;
7
8      printf("Masukkan alokasi array : ");
9      scanf("%d", &n);
10
11     // Deklarasi array sesuai alokasi yang telah ditentukan.
12     int arr[n];
13
14     // Perulangan untuk memasukkan nilai array.
15     for(i = 0; i < n; i++)
16     {
17         scanf("%d", &arr[i]);
18     }
19
20     // Keluaran.
21     for(i = 0; i < n; i++)
22     {
23         printf("Nilai array indeks ke-%d : %d\n", i, arr[i]);
24     }
25
26     return 0;
27 }
```

## Contoh Kode Program Penggunaan Array dan Implementasi For dalam Array

Program menghitung rata-rata dari nilai yang diberikan:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      // Deklarasi variabel.
6      int i, n; // Kontrol perulangan dan masukan untuk alokasi array.
7      float avg; // Nilai keseluruhan, untuk kemudian dirata-ratakan.
8
9      printf("Masukkan alokasi array : ");
10     scanf("%d", &n);
11
12     // Deklarasi array sesuai alokasi yang telah ditentukan.
13     int arr[n];
14
15     // Perulangan untuk memasukkan nilai array.
16     printf("Masukkan nilai sebanyak %d :\n", n);
17     avg = 0;
18     for(i = 0; i < n; i++)
19     {
20         scanf("%d", &arr[i]);
21         avg += arr[i];
22     }
23
24     // Keluaran.
25     printf("Rata-rata : %.2f\n", (avg / n));
26
27     return 0;
28 }
```

Program mencari nilai terkecil dan terbesar dari suatu array:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      // Deklarasi variabel.
6      int i, n,      // Kontrol perulangan dan masukan untuk alokasi array.
7          min, max; // Nilai terkecil dan terbesar.
8
9      printf("Masukkan alokasi array : ");
10     scanf("%d", &n);
11
12     // Deklarasi array sesuai alokasi yang telah ditentukan.
13     int arr[n];
14
15     // Perulangan untuk memasukkan nilai array.
16     printf("Masukkan nilai sebanyak %d :\n", n);
17     min = 99999; max = 0;
18     for(i = 0; i < n; i++)
19     {
20         scanf("%d", &arr[i]);
21
22         // Menyesuaikan nilai terkecil dan terbesar.
23         if(arr[i] < min)
24         {
25             min = arr[i];
26         }
27         if(arr[i] > max)
28         {
29             max = arr[i];
30         }
31     }
32
33     // Keluaran.
34     printf("Nilai terkecil : %d\n", min);
35     printf("Nilai terbesar : %d\n", max);
36
37     return 0;
38 }
```





## Latihan

Buatlah satu program yang terdiri dari satu array integer dinamis, lalu **tampilkan nilai minimum dan maksimum kedua** dari array tersebut.

### Contoh Masukan

```
6
1 10 31 89 34 22
```

### Contoh Keluaran

```
Nilai minimum kedua : 10
Nilai maksimum kedua : 34
```

### Contoh Masukan

```
3
5 15 7
```

### Contoh Keluaran

```
Nilai minimum kedua : 7
Nilai maksimum kedua : 7
```

---

Buatlah satu program yang terdiri dari satu array integer dinamis, lalu **tampilkan indeks nilai minimum dan maksimum berada**.

### Contoh Masukan

```
6
4 8 15 16 23 42
```

### Contoh Keluaran

```
Indeks nilai minimum : 0
Indeks nilai maksimum : 5
```

### Contoh Masukan

```
4
17 70 1 3
```

### Contoh Keluaran

```
Indeks nilai minimum : 2
Indeks nilai maksimum : 1
```



## Penutup

Terima kasih atas kerja sama seluruh pihak yang membantu dalam penyusunan modul ini, semoga apa yang telah didapatkan bisa bermanfaat di masa yang akan datang.

## Daftar Pustaka

Tim Asisten Praktikum Algoritma dan Pemrograman I Angkatan 11. (2020). Array dan Penggunaan For dalam Array. Modul Praktikum Algoritma dan Pemrograman I. Bandung: Universitas Pendidikan Indonesia.