Pertemuan 7

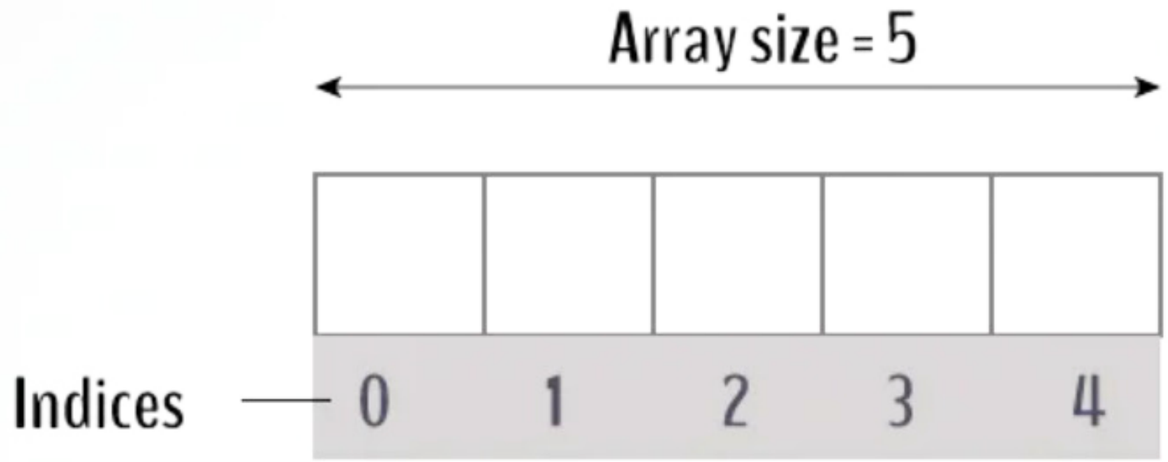
Search

Praktikum Dasar Pemrograman

Array dan Pointer

Array

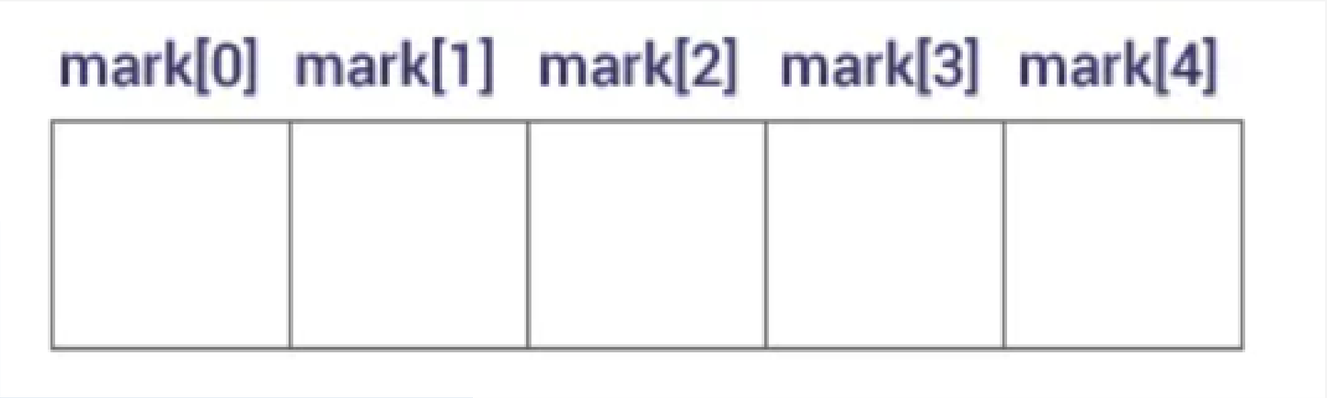
Array adalah objek yang dapat menyimpan beberapa nilai sekaligus dalam satu variabel. Bagian yang menyusun array biasa disebut **elemen array** dan tersusun secara sekuensial/berurutan. Masing-masing elemen array mempunyai **indeks** (dalam angka) sesuai dengan urutannya. Melalui indeks inilah kita dapat mengakses elemen/data yang ada di dalam array.



C Arrays

Mengakses Elemen Array

Elemen dari array dapat diakses dengan index.



Catatan Penting!

- **Index** array dimulai dari 0, bukan 1. Dari contoh di samping, marks[0] adalah elemen pertama
- Jika ukuran array adalah n, untuk mengakses elemen terakhir dari array, kita menggunakan [n-1]. Di contoh ini, marks[4]

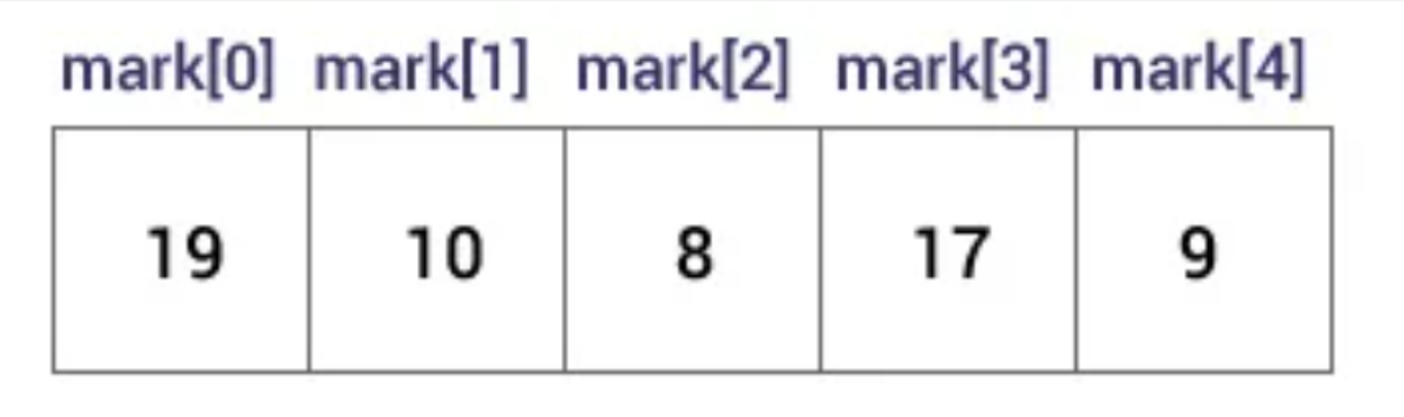
Cara Inisiasi Array

Kita bisa menginisiasi array langsung saat kita mendeklarasikannya. Misalnya:

int mark[5] = {19, 10, 8, 17, 9};

KIta juga dapat menginisiasi seperti ini:

int mark[] = {19, 10, 8, 17, 9};



Mengubah Nilai Elemen Array

```
int mark[5] = {19, 10, 8, 17, 9}

// make the value of the third element
to -1
mark[2] = -1;

// make the value of the fifth element
to 0
mark[4] = 0;
```

Input dan Output Elemen Array

Contoh untuk input array elemen:

```
// take input and store it in the 3rd element
scanf("%d", &mark[2]);

// take input and store it in the ith element
scanf("%d", &mark[i-1]);
```

Contoh untuk output array elemen:

```
// print the first element of the array
printf("%d", mark[0]);

// print the third element of the array
printf("%d", mark[2]);

// print ith element of the array
printf("%d", mark[i-1]);
```

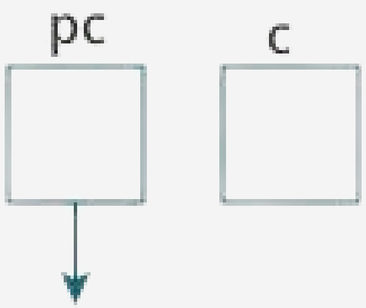
Pointer

Pointer adalah suatu variabel yang menyimpan alamat memori dari variabel lainnya. Dengan menggunakan pointer, kita dapat mengakses atau memanipulasi nilai dari variabel yang ditunjuk oleh pointer tersebut.

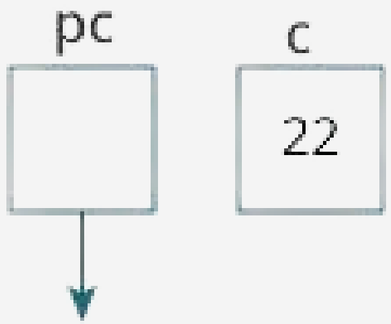
Pointer diwakili oleh tanda asterisk (*) dan digunakan dengan operator address-of (&) untuk mendapatkan **alamat memori** dari variabel. Dengan menggunakan operator dereference (*), kita dapat mengakses nilai dari variabel yang ditunjuk oleh pointer tersebut.

contoh5_5.c

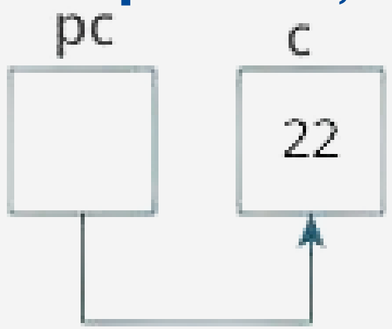
1. int* pc, c;



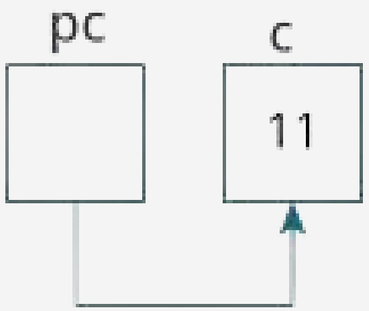
2. c = 22;



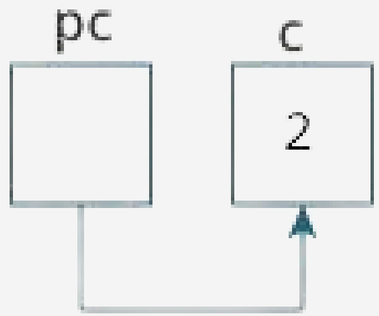
3. pc = &c;



4. c = 11;



5. *pc = 2;



Thank you.

Any question?