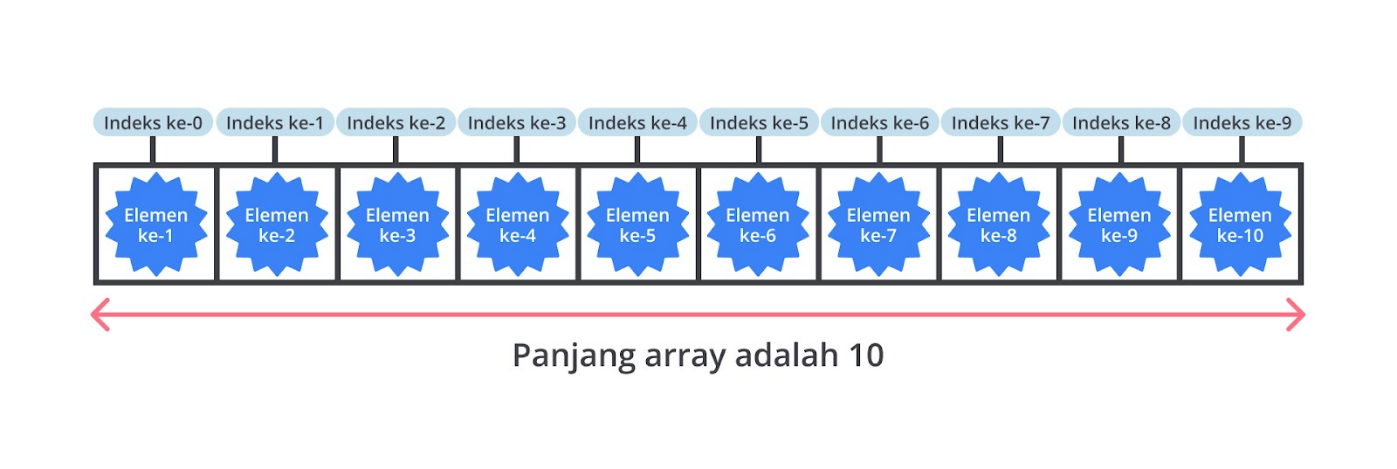
**Rangkuman Array dan Pemrosesannya**

Kita sudah berada di penghujung materi Array dan Pemrosesannya. Sampai sini, Anda telah mempelajari salah satu struktur data, yakni array. Mari kita rangkum secara saksama.

**Fundamental Array**

Array adalah salah satu jenis dari struktur data linear dan terdiri dari kumpulan elemen bertipe data sama dengan indeks yang berurutan atau linear. Struktur array adalah berikut.



Berikut adalah penjelasan dari struktur tersebut.

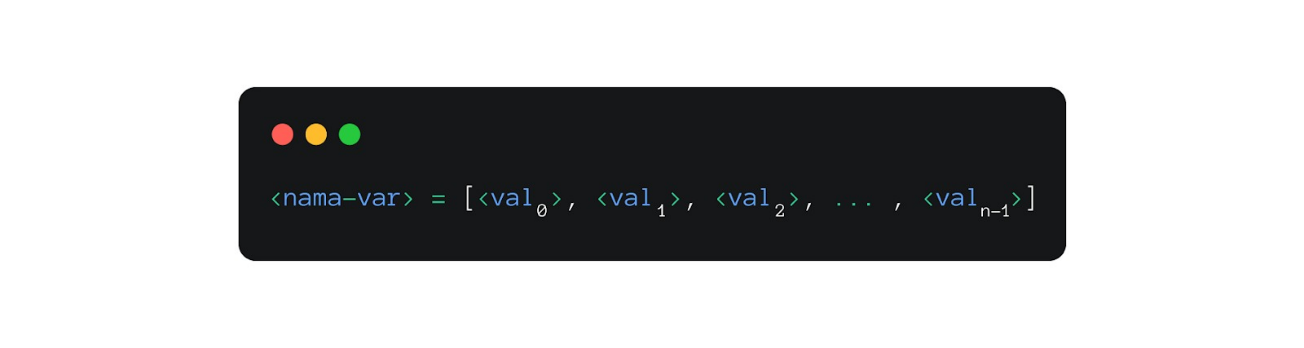
1. Indeks-indeks (indices): Kumpulan indeks yang keseluruhan dari kumpulan indeks tersebut disebut sebagai array.
2. Indeks pertama: Indeks pertama array yang selalu dimulai dari 0.
3. Element: Nilai yang berada dalam suatu indeks, contohnya jika nilai dari indeks 8 adalah string "Dicoding", kita bisa sebut sebagai "elemen ke-9 adalah string 'Dicoding'".
4. Array length: Panjang dari suatu array. Dalam gambar tersebut, panjang array adalah 10.

Dalam Python terdapat dua cara untuk menggunakan array, yakni berikut.

1. Menggunakan List
   1. x = [1, 2, 3, 4, 5]
   2. print(x)
   4. """
   5. Output:
   6. [1, 2, 3, 4, 5]
   7. """
2. Menggunakan Library Array
   1. import array
   3. x = array.array("i",[1, 2, 3, 4, 5])
   4. print(x)
   5. print(type(x))
   7. """
   8. Output:
   9. array('i', [1, 2, 3, 4, 5])
   10. <class 'array.array'>
   11. """

**Implementasi Array dengan Python**

Dalam kelas ini, kita telah mempelajari array menggunakan list Python. Secara detail, ada dua cara untuk melakukan deklarasi array menggunakan list Python, yakni berikut.

1. **Mendefinisikan Isi Array**  
   Cara pertama adalah dengan mendeklarasikan variabel array sekaligus mendefinisikan isi array. Cara ini dilakukan jika kita sudah tahu nilai yang perlu diberikan.  
     
   Berikut adalah struktur mendeklarasikan variabel array dengan mendefinisikan isi array secara langsung.  
     
   Dari struktur tersebut, **<nama-var>**merupakan nama variabel array yang dideklarasikan sebanyak **n** dengan elemen-elemennya adalah **<val0>, <val1>, <val2>, … , <valn-1>**. Perlu diingat bahwa elemen tersebut terurut berdasarkan indeks dari **0** hingga **n-1**.
2. **Mendefinisikan Nilai Default**  
   Cara kedua adalah mendeklarasikan array dengan mendefinisikan nilai default. Cara ini dilakukan jika kita tidak mengetahui nilai yang diberikan. Kita dapat memberikan nilai default terlebih dahulu sebagai upaya untuk memberikan nilai awal.  
     
   Nilai default ditentukan oleh kesepakatan bersama sesuai kebutuhan yang nilainya di luar dari rentang yang ditentukan. Misalnya tim Anda menentukan nilai dalam list harus berkisar dari 1 hingga 10. Kita bisa menyepakati "0" sebagai nilai default karena di luar jangkauan yang disepakati (1-10).  
     
   Berikut adalah struktur mendeklarasikan variabel array dengan mendefinisikan nilai default.  
   [](https://www.dicoding.com/academies/86/tutorials/5084)  
   Berikut adalah penjelasan lebih detail terkait struktur tersebut.
   1. <nama-var> merupakan variabel yang Anda deklarasikan.
   2. <default-val> merupakan nilai default yang Anda definisikan. Umumnya, programmer akan menggunakan nilai di luar range yang telah disepakati sebagai nilai default. Misalnya jika range nilai yang disepakati seharusnya 1 hingga 10, nilai default bisa kita definisikan dengan 0.
   3. <n> merupakan ukuran panjangnya array.

**Pemrosesan Sekuensial pada Array**

Pemrosesan array merujuk pada operasi-operasi yang dilakukan pada elemen-elemen suatu array. Operasi ini melibatkan manipulasi hingga pengolahan elemen yang ada pada array. Adapun pemrosesan sekuensial adalah sebuah pemrosesan setiap elemen array yang dimulai dari elemen pada indeks terkecil hingga terbesar. Pemrosesan sekuensial lebih sering menggunakan pengulangan (loop/iterasi) dalam setiap prosesnya.

Sebab pemrosesan sekuensial melibatkan semua elemen di dalamnya, ada beberapa hal yang perlu diperhatikan.

1. Setiap elemen array diakses secara langsung melalui indeksnya (metode indexing).
2. Elemen pertama (*first element*) adalah elemen array dengan indeks terkecil yang selalu dimulai dari 0.
3. Elemen selanjutnya (*next element*) dicapai melalui suksesor indeks.
4. Kondisi berhenti dicapai jika indeks yang diproses adalah indeks terbesar yang sudah terdefinisi.
5. Suatu array tidak boleh kosong, minimal memiliki **satu** elemen di dalamnya.

Banyak sekali contoh penerapan pemrosesan sekuensial pada array, beberapa di antaranya sebagai berikut.

1. Mengisi array secara sekuensial.
2. Menghitung nilai rata-rata elemen array.
3. Mengalikan elemen array dengan suatu nilai.
4. Mencari nilai terbesar atau terkecil pada array.
5. Mencari indeks letak suatu nilai ditemukan pertama kali dalam array, dan sebagainya.