**STRUKTUR DATA: ARRAY DUA DIMENSI**

1. **Definisi**

Array dua dimensi adalah array yang memiliki array di dalamnya atau dapat disebut array di dalam array.

Contohnya:

A = [

[1,2,3],

[3,4,5]

];

Dimana A[2][3], [2] menandakan baris dan [3] menandakan kolom. Salah satu implementasinya yaitu pada matriks, jika diberikan Matriks[3][3]. Berarti array matriks akan berdimensi 3 x 3.

Matriks = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9],

];

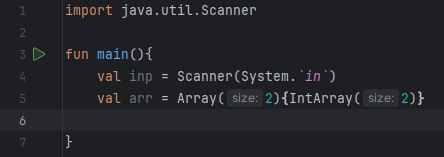
Array memiliki indeks yang dimulai dari 0, untuk lebih mudah mengetahui indeks per elemennya kita bisa buatkan table.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | M[0][0] | M[0][1] | M[0][2] |
| 1 | M[1][0] | M[1][1] | M[1][2] |
| 2 | M[2][0] | M[2][1] | M[2][2] |

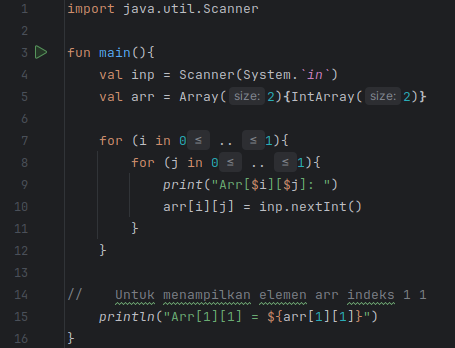
Kita dapat memanggil isi array nya dengan memanggil indeks array-nya, misal M[0][2].

1. **Implementasi Pada Program Kotlin**

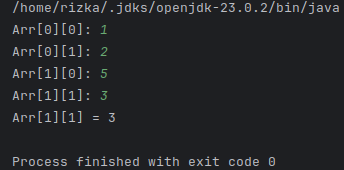
Mendeklarasikan array dua dimensi berukuran 2 baris 2 kolom.



Membuat program dimana isi array diinput oleh user

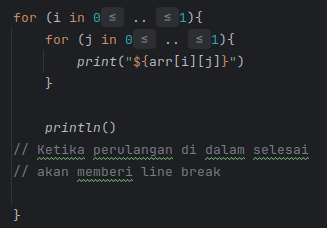


Kita menggunakan perulangan bercabang dimana pada perulangan pertama i akan mewakili baris dan perulangan kedua j akan mewakili kolom. Kemudian kita meminta user menginput nilai per-elemen dan menampungnya dalam array sesuai indeks masing-masing.

Hasilnya, user akan diminta menginput isi array lalu memanggil elemen arr indeks [1][1]:

Menampilkan seluruh isi array dengan beberapa cara:

* + 1. For Loop dengan nilai akhir yang sudah ditetapkan

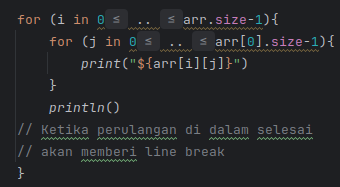
Menggunakan perulangan bercabang dan nilai akhir dari perulangan sudah ditetapkan 1, hal ini karena panjang array adalah 2. Cara ini tidak efisien karena tidak bisa mengubah panjang perulangan dengan dinamis. Jika suatu saat kita mengubah panjang array maka harus mengubah nilai akhir pada perulangan juga, tentu kurang efisien. Jadi diberikan cara lain lagi yang lebih dinamis.

* + 1. Menggunakan *array.size*

Pada kotlin, kita diberi sebuah method yang dapat mengetahui ukuran dari array yaitu *size.*

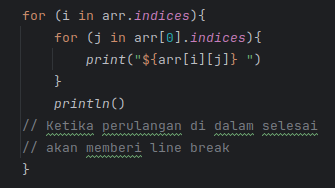
Hasilnya:

Implementasikan pada perulangan bersarang untuk menampilkan array.



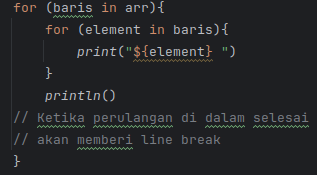
* + 1. Menggunakan array.indicies

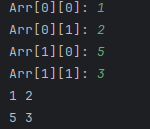
Terdapat cara yang lebih efisien lagi untuk mengetahui panjang dari array. Jika array.size mengembalikan panjang array, array.indices mengembalikan panjang indeks dari array sehingga tidak perlu dikurangi 1.



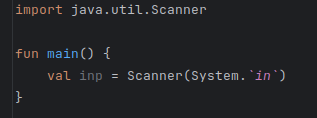
* + 1. Tanpa menggunakan indeks

Jika sebelumnya kita menampilkan array per-indeks elemennya, sekarang kita membuat variabel yang menampung isi dari array langsung.

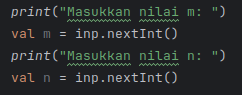
Pada perulangan pertama, variabel baris akan menampung array yang ada dalam array kemudian perulangan selanjutnya menampung masing-masing elemen array. Sehingga kita hanya perlu memanggil element untuk menampilkan isi array. Variabel baris dan element bukan nama tetap, jadi bisa diganti sesuka hati.

Hasil dari cara-cara diatas sama, yaitu:

1. **Studi kasus perhitungan dua matriks (m\*A) + (n\*B)**
   1. Membuat program di mana isi array dapat diinput oleh user, kita harus membuat variabel untuk menampung Scanner.

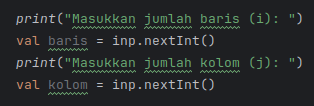
Kita bisa membuat variabel *inp* menggunakan *val* untuk menampung *Scanner* tersebut.

* 1. Minta user menginput nilai m dan n dan simpan inputan ke dalam sebuah variabel



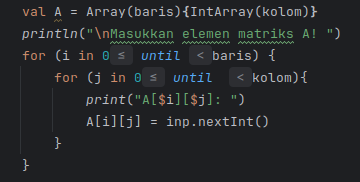
Output:

* 1. Minta user menginput jumlah baris dan kolom, lalu masukkan dalam variabel baris dan kolom

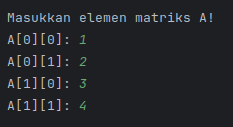


Output:

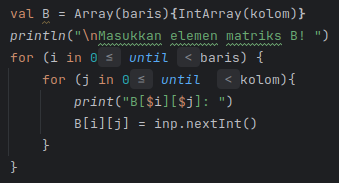
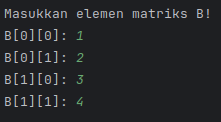
* 1. Buat matriks A dengan panjang baris dan kolom sesuai dengan yang diinput oleh user



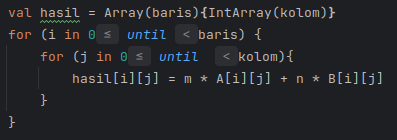
Pertama deklarasikan array A dengan ukuran baris dan kolom, buat perulangan bercabang lalu minta user menginput elemen-elemen array dan masukkan inputan itu ke dalam array.

Output:

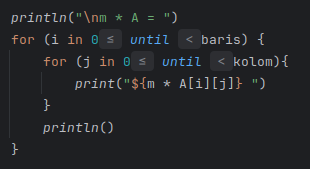
* 1. Lakukan hal yang pada matriks B

Output:

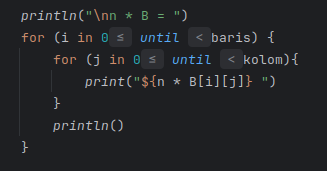
* 1. Menghitung m\*A + n\*B dan memasukkan hasilnya ke dalam array baru yaitu array hasil



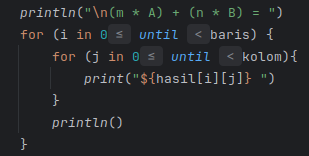
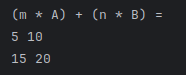
* 1. Menampilkan hasil m dikali matriks A ke layar

Output:

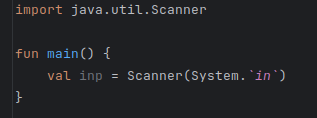
* 1. Menampilkan hasil n dikali matriks B ke layar

Output:

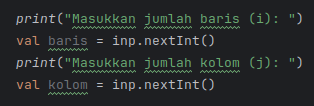
* 1. Menampilkan matriks hasil ke layar

Output:

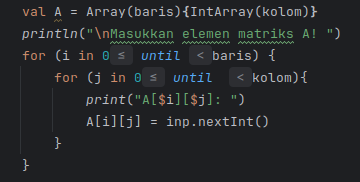
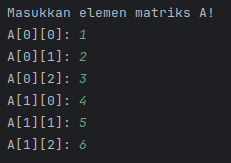
1. **Studi Kasus Matriks Transpose**
   1. Membuat program di mana isi array dapat diinput oleh user, kita harus membuat variabel untuk menampung Scanner.



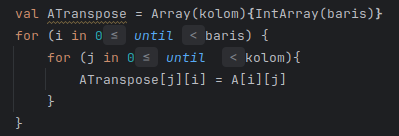
* 1. Minta user menginput jumlah baris dan kolom, lalu masukkan dalam variabel baris dan kolom

Output:

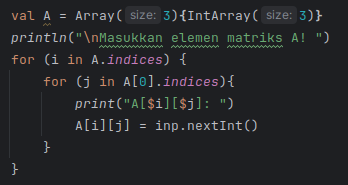
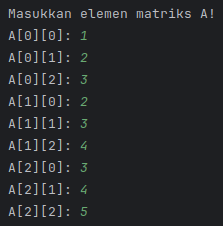
* 1. Buat matriks A dengan panjang baris dan kolom sesuai dengan yang diinput oleh user

  Output:

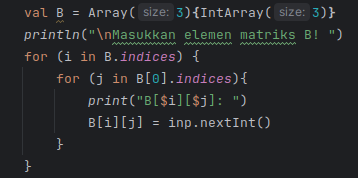
* 1. Membuat matriks ATranspose dengan menukar baris menjadi kolom dan kolom menjadi baris

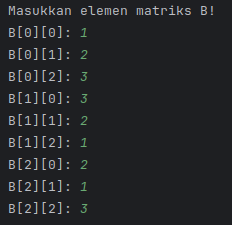


1. Studi Kasus di Kelas
   1. Buatlah program perhitungan penjumlahan dan pengurangan 2 buah matriks A dan B dengan ukuran matrik 3 x 3
      * Karena ukuran sudah ditentukan, maka tidak perlu meminta user menginput jumlah baris dan kolomnya
      * Saat meminta input elemen array, saya menggunakan array.indices agar lebih efisien
      * Minta user menginput elemen-elemen matriks A

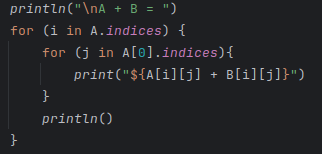
Output:

* + - Minta user menginput elemen-elemen matriks B

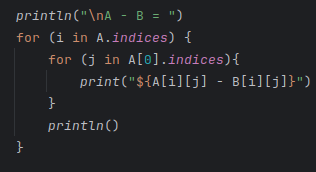


Output:

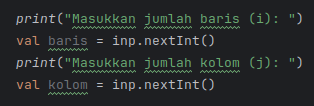
* + - Tampilkan hasil penjumlahan matriks A dan B

Output:

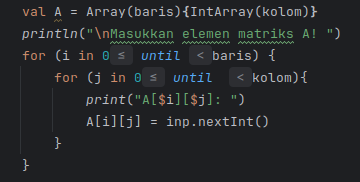
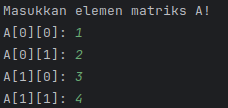
* + - Tampilkan hasil pengurangan matriks A dan B

Output:

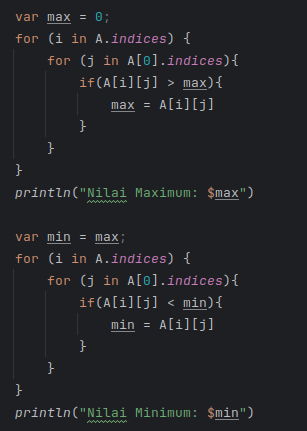
* 1. Buatlah program pencarian nilai minimum dan maximum pada matriks
  + Minta user menginput jumlah baris dan kolom, lalu masukkan dalam variabel baris dan kolom

Output:

* + Buat matriks A dengan panjang baris dan kolom sesuai dengan yang diinput oleh user

 Output:

* + Tampilkan nilai maximum dan minimum



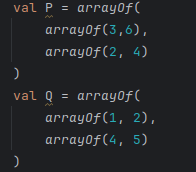
Output:

Pertama buatlah sebuah variabel max yang menyimpan nilai awal 0 kemudian bandingkan dengan masing-masing elemen array, jika elemen array lebih besar dari nilai max maka nilai max akan menyimpan elemen itu hingga didapatlah nilai akhir yang terbesar. Sedangkan untuk nilai min akan menyimpan nilai max dahulu dan membandingkan pada masing-masing elemen array, jika lebih kecil dari nilai min maka nilai min akan menyimpan elemen tersebut hingga didapatlah nilai terkecil.

* 1. Buatlah program perkalian matriks ilustrasi

P = Q =

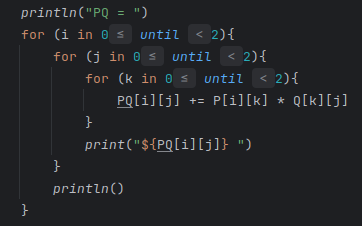
* + - Buat array P dan Q



* + - Deklarasikan array PQ yang menampung hasil perkalian P dan Q



* + - Menggunakan perulangan bercabang untuk memasukan hasil perkalian P dan Q ke dalam matriks PQ, kemudian menampilkan ke layar



Output: