**LAPORAN PRAKTIKUM PENGOLAHAN CITRA DIGITAL**

**1. MATLAB DATA STRUCTURES**



**Disusun oleh :**

**Nama : Felix**

**NPM : 2327250059**

**Kelas : IF41**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**2025**

**TUTORIAL 1. MATLAB DATA STRUCTURES**

**Goal**

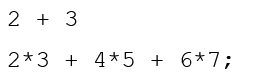
The goal of this tutorial is to learn how to create, initialize, and access some of the most useful data structures available in MATLAB.

**Objectives**

* Learn how to use MATLAB for basic calculations involving variables, arrays, matrices, and vectors.
* Explore multidimensional and cell arrays.
* Review matrix operations.
* Learn how to use MATLAB structures.
* Explore useful functions that can be used with MATLAB data structures.

**Procedure**

1. Execute the following lines of code one at a time in the *Command Window* to see how MATLAB can be used as a calculator.



|  |  |  |
| --- | --- | --- |
|  |  |  |

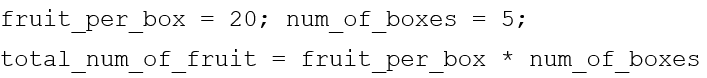
***Question 1*** What is the variable **ans** used for?

|  |  |
| --- | --- |
| Ans adalah sebuah variabel yang dibuat oleh mathlab agar mempermudah kita melihat hasil akhir, akan tetapi ans tersebut muncul karena kita tidak menyebutkan spesifik hasil outputnya, bisa di ubah dengan cara seperti b = 2 + 3 akan menghasilkan b = 2 + 3.   |  | | --- | |  | |

***Question 2*** What is the purpose of using a semicolon (;) at the end of a statement?

|  |  |
| --- | --- |
|  |  |
| Penggunaan titik koma (;) menurut saya setelah mencoba bebrapa code di matlab saya menyimpilkan bahwa agar suatu *code* yang di ketikkan pada matlab tidak lengsung di jalankan, melainkan di simpan contohnya a = 2 ; b = 5; c = a \* b outputnya c = 10. | Sedangkan dengan tidak menggunakan semicolom(;) langsung menampilkan hasilnya |

1. Perform calculations using variables.

******

|  |  |
| --- | --- |
|  |  |

***Question 3*** Experiment with creating your own variables. Are variables case sensitive?

|  |  |
| --- | --- |
|  |  |
| Ya pada matlab mereka menggunakan case sensitive, maka seperti pada baris ke 2 hasilnya error. | |

***Question 4*** What is the value/purpose of these variables: pi, eps, inf, i? Is it possible to overwrite any of these variables? If so, how can it be undone?

|  |  |
| --- | --- |
|  |  |
|  | untuk menjawab apakah variable-variabel tersebut bisa di timpa/ diubah? Ya bisa dengan menggunakan penamaan variable yang sama saja lalu diubah, contoh pi = 5 maka hasil Ketika kita ketik pi maka akan menghasilkan outpun pi = 5.  Dan bagaimana cara kita mengembalikannnya lagi seperti semula dengan menggetikkan code “clear <nama variable>” dan “clear all” akan tetapi penggunakan kode yang ke-2 harus lebih hati-hati, karena menggunakan kode tersebut akan membuat seluruh variable yang kita buat sebelumnya kan hilang semuanya. |

1. Execute the commands who and whos, one at a time, to see their function and the difference between them.

There are several commands that will keep the MATLAB environment clean. Use them whenever you feel your command window or workspace is cluttered with statements and variables.

1. Clear a variable in the workspace. After execution, note how the variable disappears from the workspace.



1. Clear the command window and all variables with the following lines of code (one at a time to see their effects individually).



1. Create a 3 × 3 matrix by executing the following line of code.



|  |  |
| --- | --- |
|  |  |

***Question 5*** What is the use of the semicolon in this statement?

|  |  |
| --- | --- |
|  |  |
| Dari saya tes pada matlab selain untuk agar code tidak lengsung di jalankan titik koma (*semicolon*) berfungsi juga untuk membuat matriks akhir dari titik koma maka tandanya akhir dari baris pertama dan seterusnya. | |

**The Colon Operator**

1. A very useful operator in MATLAB is the colon (:). It can be used to create a vector of numbers.



|  |
| --- |
|  |

1. A third parameter determines how to count between the starting and ending numbers. It is specified between the start and end values.



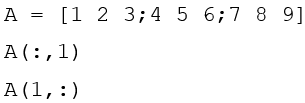


|  |  |
| --- | --- |
|  |  |

***Question 6*** Write a statement that will generate a vector of values ranging from 0 to *π* in increments of *π/*4.

|  |  |
| --- | --- |
|  |  |

1. The colon operator can also be used to return an entire row or column of a matrix.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

***Question 7*** Write a line of code that would generate the same 3×3 matrix as in the variable A above, but using the colon operator to generate the sequence of numbers in each row instead of explicitly writing them.

|  |  |
| --- | --- |
|  |  |
|  |  |

1. The colon operator can be replaced with the function colon, which performs the same operation.



|  |  |
| --- | --- |
|  |  |

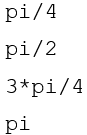
As seen in the steps above, creating a vector of evenly spaced numbers is easily done with the colon (:) operator when we know where the vector starts, ends, and how large the space in between each value is. In certain cases, we may wish to create a vector of numbers that range between two numbers, but we only know the quantity of values needed (for example, create a vector that consists of 4 values between *π/*4 and *π*) . To do this, we use the linspace function.

1. Execute this command to see how the function linspace operates.



|  |  |
| --- | --- |
|  |  |

1. Compare the result from the previous step with these values.

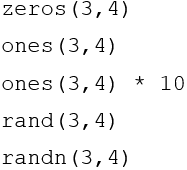


|  |  |
| --- | --- |
|  |  |

**Special Built-In Matrices**

MATLAB has several built-in functions that will generate frequently used matrices automatically.

1. Execute the following lines of code one at a time.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***Question 8*** What is the difference between the functions rand(M,N) and randn(M,N)?

|  |  |
| --- | --- |
|  |  |
|  |  |
| Rand menggunakan rentang [0,1] sedangkan randn menggunakan rentang[-1,1] | |

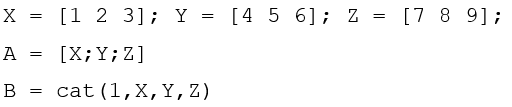
**Matrix Concatenation**

Concatenation of matrices is done with brackets ([]) or using the cat function. Take, for example, the statement

A = [1 2 3;4 5 6;7 8 9]

The brackets are combining three rows. Instead of explicitly defining each row all at once, they can be defined individually as vectors and then combined into a matrix using brackets.

1. Combine the three individual vectors into a 3 × 3 matrix.



|  |  |
| --- | --- |
|  |  |

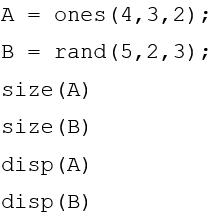
Similarly, the brackets can be used to delete a row of a matrix.

1. Delete the last row (row 3) of the matrix A. Note how the colon operator is used to specify the entire row.

A(3,:) = []

A vector with *N* elements is an array with one row and *N* columns. An element of a vector can be accessed easily by addressing the number of the element, as in X(5), which would access the fifth element of vector X. An element of a two-dimensional matrix is accessed by first specifying the row, then the column, as in X(2,5), which would return the element at row 2, column 5. Matrices of dimensions higher than 2 can be accessed in a similar fashion. It is relevant to note that arrays in MATLAB are 1-based—the first element of an array is assigned or accessed using 1, as opposed to 0, which is the standard in many programming languages.

1. Use the ones and rand functions to create multidimensional arrays.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

***Question 9*** What does the size function do?

|  |  |
| --- | --- |
|  |  |
| Size() pada matlab berfungsi untuk melihat ukuran suatu array atau yang lain, akan tatapi menggembalikannya dalam bentuk ukuran matriks. Contohnya seperti A = ones(4,3,2); size(A) maka akan tampil ans = 4 3 2 itu menunjukkan bahwa variable A berukuran matriks 4x3x2 atau dengan kata lain 3 dimensi. | |

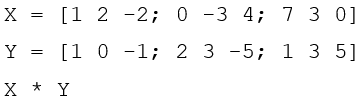
***Question 10*** What does the disp function do?

|  |  |
| --- | --- |
|  |  |
| Disp() pada matlab berfungsi untuk menampilkan seluruh isi pada variabel. | |

**Operations Involving Matrices**

Performing arithmetic operations on matrices can be achieved with the operators + - \* /. The default for the multiply (\*) and divide (/) operators is matrix multiplication and matrix division. To perform arithmetic operations on individual elements of a matrix, precede the operator with a dot (.).

1. Perform matrix multiplication on two matrices.



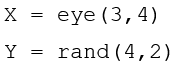
|  |  |
| --- | --- |
|  |  |
|  |  |

1. Perform element-by-element multiplication.



|  |  |
| --- | --- |
|  |  |

1. Perform another matrix multiplication on two matrices.



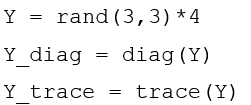


|  |  |
| --- | --- |
|  |  |
|  | ***X\*Y***  ***y\*x error*** |

***Question 11*** Why did the last operation fail?

|  |
| --- |
| Dikarenakan y memiliki 4 kolom dan x meiliki 3 kolom, maka dari itu baris terakhir dari code tersebut tidak bisa atau error, pada matriks juga seperti itu. |

1. Use the diag and trace functions to perform operations on the diagonal of a matrix.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

***Question 12*** What does the diag function do?

|  |
| --- |
| Diag() berfungsi untuk mengambil nilai diagonal pada sebuah matriks lalu di tampilkan. |

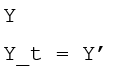
***Question 13*** What does the trace function do?

|  |
| --- |
| Trace() berfungsi untuk menjumlahkan dari nilai-nilai diagonal dari matriks. |

***Question 14*** Write an alternative statement that would produce the same results as the trace function.

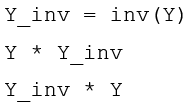
|  |  |  |
| --- | --- | --- |
|  |  | Hanya bisa matriks 3\*3. |
|  |  | Ini bisa semua matrik. |

1. Calculate the transpose of a matrix.



|  |  |
| --- | --- |
|  |  |

1. Calculate the inverse of a matrix and show that *YY* −1 = *Y*−1*Y* = *I*, where *I* isthe identity matrix.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

1. Calculate the determinant of a matrix.

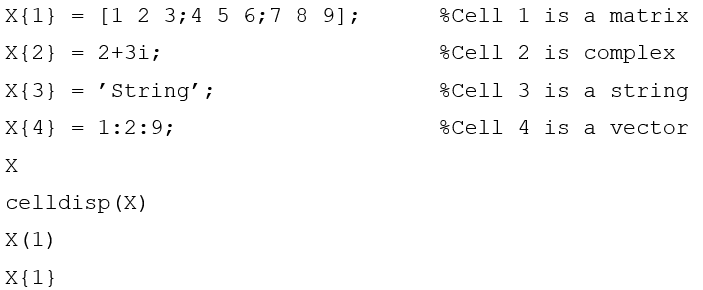
****

|  |  |
| --- | --- |
|  |  |

**Cell Array**

As demonstrated earlier, a matrix is the fundamental data type in MATLAB. It resembles the classical definition of an array as a homogeneous data structure, that is, one in which all its components are of the same type. *Cell arrays*, on the other hand, are another type of array where each cell can be any data type allowed by MATLAB. Each cell is independent of another and, therefore, can contain any data type that MATLAB supports. When using cell arrays, one must be careful when accessing or assigning a value to a cell; instead of parentheses, curly braces ({}) must be used.

1. Execute the following lines of code one at a time to see how cell arrays are handled in MATLAB.

******

|  |  |
| --- | --- |
|  | *Menyimpan pada variable x* |
|  |  |
|  |  |
|  |  |
|  |  |

***Question 15*** What does the celldisp function do?

|  |  |
| --- | --- |
|  |  |
| *Celldisp() berfungsi untuk menampilkan seluruh isi dari x.* | |

***Question 16*** What does the percent (%) character do?

|  |
| --- |
|  |
| Fungsi percent(%) adalah untuk komentar suatu code |

***Question 17*** What is the difference between the last two lines in the code above (X(1) as opposed to X{1})?

|  |  |  |
| --- | --- | --- |
|  |  | *Fungsi x(1) hanya memberitahukan informasi, pada contoh mengeluarkan informasi isinya apa.* |
|  |  | *Sedangkan x{1} berfungsi untuk memberitahukan lalu membuka apa isinya pada index ke-1.* |

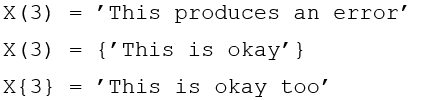
There is another way to assign values to a cell array that is syntactically different, but yields the same results. Note in the next step how the cell index is enclosed within normal parentheses (()), but the data that will be saved to the cell is encapsulated by curly braces ({}).

1. Execute this line to see another way of assigning cell array values.



|  |  |
| --- | --- |
|  |  |

1. The next few lines of code will demonstrate proper and improper ways of cell array assignment when dealing with strings.

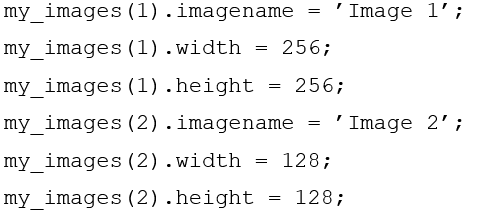
****

|  |  |
| --- | --- |
|  | Error karena tipe data yang di gunakan berbeda dari sebelumnya. |
|  | Kode ini bisa di jalankan karena memenuhi syarat dan juga dengan tipe data yang sama. |
|  | Kode ini bisa, karena kode tersebut langung menuju ke isi indeksnya maka bisa bekerja. |

**Structures**

Structures are yet another way of storing data in MATLAB. The syntax for structures is similar to that of other programming languages. We use the dot (.) operator to refer to different fields in a structure. Structures with identical layout (number of fields, their names, size, and meaning) can be combined in an array (of structures).

1. Create an array of two structures that represents two images and their sizes.



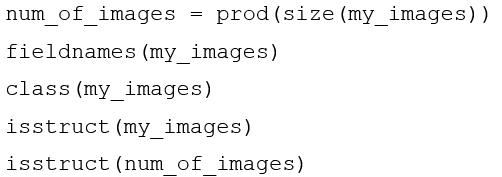
|  |  |
| --- | --- |
|  |  |
|  |  |

1. View details about the structure and display the contents of a field.



|  |  |
| --- | --- |
|  |  |
|  |  |

1. Display information about the structure.

******

|  |
| --- |
|  |

***Question 18*** What does the fieldnames function do?

|  |  |  |
| --- | --- | --- |
|  |  | Fungsi dari fieldnames() adalah untuk mengecek ada struktur apa saja di dalam my\_images |

***Question 19*** What does it mean when the result from the function isstruct is 1? What does it mean when it is 0?

|  |  |  |
| --- | --- | --- |
|  | my\_images    num\_of\_images | Bila hasilnya 1 maka benar bahwa variable tersebut merupakan Struct, bila 0 maka salah. |

***Question 20*** Use the help system to determine what function can be used to delete a field from a structure.

|  |  |  |
| --- | --- | --- |
|  |  | maka akan menghapus width pada my\_images. |