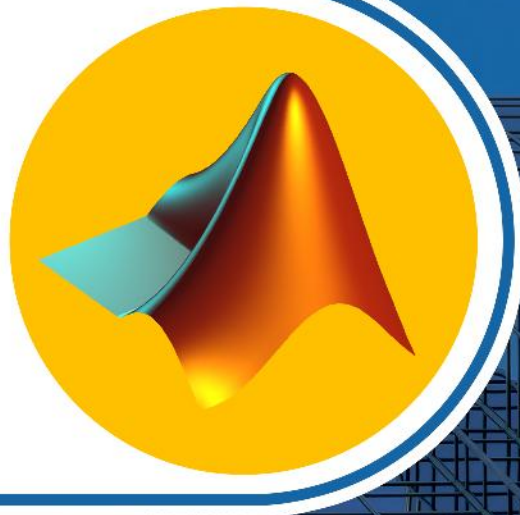




KONSEP

SEGMENTASI DAN KLASIFIKASI CITRA MEDIS DISERTAI KODE PROGRAM MATLAB

Arif Muntasa
Devie Rosa Anamisa



Konsep Segmentasi dan Klasifikasi Citra Medis (Disertai Kode Program Matlab)

Arif Muntasa
Devie Rosa Anamisa



Konsep Segmentasi dan Klasifikasi Citra Medis (Disertai Kode Program Matlab)

Penulis:
Arif Muntasa
Devie Rosa Anamisa

Editor:
Erik Santoso

Layouter :
Tim Kreatif PRCI

Cover:
Rusli

Cetakan Pertama : November 2022

Hak Cipta 2022, pada Penulis. Diterbitkan pertama kali oleh:

**Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT**

Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org
E-mail : rumahcemerlangindonesia@gmail.com

Copyright © 2022 by Perkumpulan Rumah Cemerlang Indonesia
All Right Reserved

- Cet. I – : Perkumpulan Rumah Cemerlang Indonesia, 2022
; 15,5 x 23 cm
ISBN : 978-623-448-273-7

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

KATA PENGANTAR

Dengan diselesaikannya Buku ini, maka diharapkan mahasiswa yang sedang tugas akhir dengan topik citra medis sebagai dataset untuk eksperimen tidak ada lagi kesulitan. Karena dalam buku ini telah dibahas konsep dan implementasinya mulai dari citra digital, segmentasi citra medis dengan tahapan yang cukup Panjang, namun tetap mudah untuk dipahami. Dalam buku ini, kami menggunakan Matlab sebagai High Level Language yang sangat mudah untuk dipahami, meskipun kecepatan proses awal diperlukan memory cukup besar, karena semua library dipanggil saat pertama kali memanggil Matlab. Namun dibalik beban berat pada saat pemanggilan awal, Matlab sangat mudah digunakan dan telah banyak Library yang siap digunakan.

Bagimana juga buku ini masih banyak kekurangan. Kami akan terus selalu menulis dan memperbaiki kekurangan- kekurangan. Kami siap menerima saran yang sifatnya adalah untuk membangun demi kebaikan saya secara pribadi dan perbaikan dari buku yang kami tulis

Terimakasih

Surabaya, November 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	I
DAFTAR ISI	II
BAB 1. PENGENALAN MATLAB	1
1.1 Current folder pada Matlab	1
1.2. Command window pada Matlab	2
1.3. Workspace pada Matlab	3
1.4. Editor	5
1.5. Menu	6
1.6. Penggunaan <i>set path</i> untuk memudahkan eksperimen	7
1.7. Mendebug program menggunakan matlab	9
BAB 2. OPERATOR PADA MATLAB	13
2.1. Operator aritmatika	13
2.1.1 Operator penjumlahan pada matlab	15
2.1.2 Operator pengurangan pada Matlab	16
2.1.3 Operator perkalian dan perkalian antar elemen pada matlab	17
2.1.4 operator pembagian pada Matlab	20
2.2. Operator relasional	22
2.3. Operator logical	22
BAB 3. TIPE DATA PADA MATLAB	27
3.1 Array pada Matlab	30
3.1.1. Array satu dimensi	31
3.1.2. Array dua dimensi	32
3.1.3. Lebih jauh tentang array tiga dimensi atau dimensi yang lebih dari tiga	48
BAB 4. PENGOLAHAN CITRA DIGITAL	49
4.1. Citra True Colour	51
4.2. Citra Grayscale	54
4.3. Citra HSV	57
BAB 5. SEGMENTASI CITRA MEDIS	59
5.1. Pemisahan <i>channel</i> merah, hijau, dan biru dari citra <i>true color</i>	64
5.2. Memperbaiki <i>channel</i> hasil pemisahan	65
5.3. Menggabungkan Kembali hasil perbaikan	66
5.4. Melakukan konversi ke citra HSV	68
5.5. Memeilih channel Hue (H) untuk digunakan sebagai acuan dalam memperoleh nilai threshold	70

5.6. Menghitung nilai threshold menggunakan metode Otsu	71
5.7. Mengubah citra <i>Hue</i> menjadi citra biner	73
5.8. Melakukan dilasi hasil binerisasi dan diikuti dengan proses erosi, atau sering disebut dengan operasi closing dari operasi morfologi.	74
5.9 Menempelkan hasil segmentasi pada citra asli, sehingga diperoleh hasil segmentasi tetap citra <i>true colour</i> .	75
BAB 6 PEMBENTUKAN GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)	77
BAB – 7. KLASIFIKASI BERBASIS JARAK	83
7.1. Euclidean Distance Similarity	84
7.2. Manhattan Similarity	91
7.3. Minkowski Distance Similarity	96
BAB 8 - DASAR-DASAR CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI	109
8.1. Convolutional neural network	109
8.2. Mengenal Arsitektur convolutional neural network	120
8.3. Implementasi CNN Arsitektur AlexNet Untuk Klasifikasi Citra Leukemia Menggunakan Matlab	127
DAFTAR PUSTAKA	131

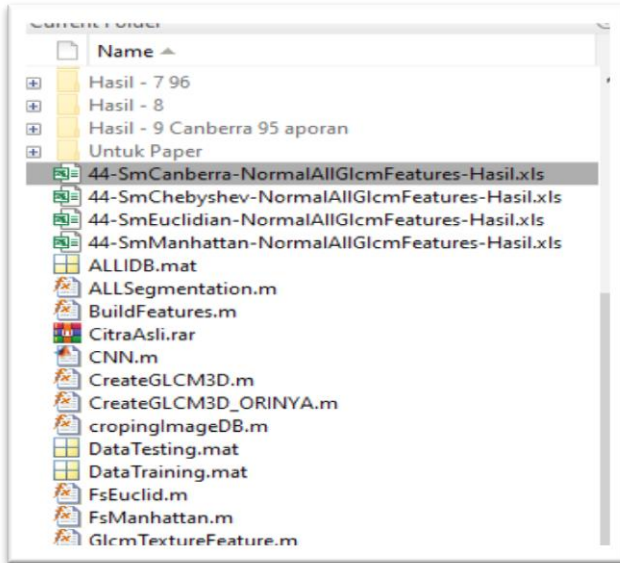
Bab 1. Pengenalan Matlab

Matlab merupakan salah satu bahasa pemrograman yang banyak digunakan oleh peneliti untuk melakukan eksperimen. Sebelum membuat program menggunakan Matlab, kami tunjukkan terlebih dahulu *interface* yang ada pada matlab. Ada beberapa bagian yang perlu diketahui, diantaranya adalah

1. Current Folder
2. Command Window
3. Workspace
4. Editor
5. Menu

1.1 Current folder pada Matlab

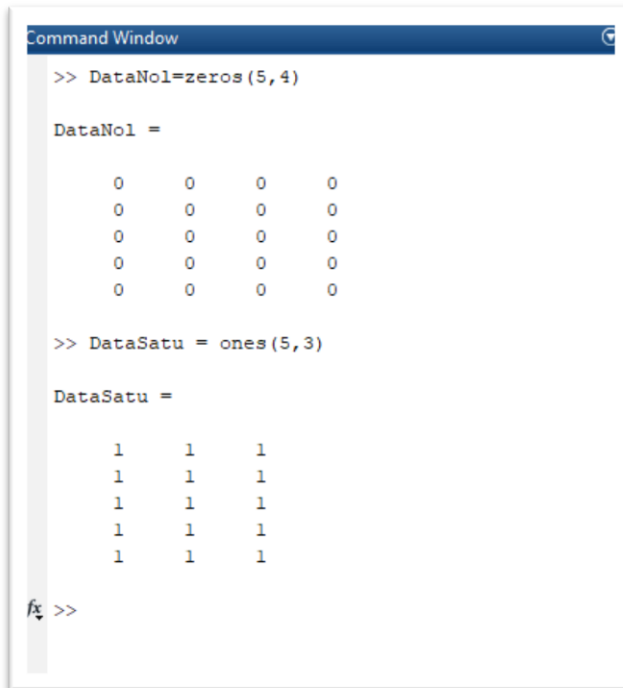
Current folder merupakan folder yang kita pilih untuk digunakan sebagai tempat penyimpanan program, data, atau pun citra sebagaimana ditunjukkan pada Gambar 1.1. Default dari *current folder* biasanya adalah lokasi aplikasi matlab disimpan, oleh karena itu perlu kita memilih folder yang kita butuhkan.



Gambar 1.1. Interface Pada Matlab

1.2. Command window pada Matlab

Salah satu keunggulan dari Matlab adalah fasilitas command window, sehingga memungkinkan kita untuk mencoba terlebih dahulu perintah matlab sebelum kita gunakan pada program, sebagaimana ditunjukkan pada Gambar 1.2. Pada command window juga kita dapat melihat apakah perintah yang kita gunakan benar atau salah. Jika benar, maka akan memberikan respon hasil. Jika perintah kita salah, maka Matlab akan menampilkan informasi kesalahan yang kita lakukan.



```
>> DataNol=zeros(5,4)

DataNol =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0

>> DataSatu = ones(5,3)

DataSatu =

     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1

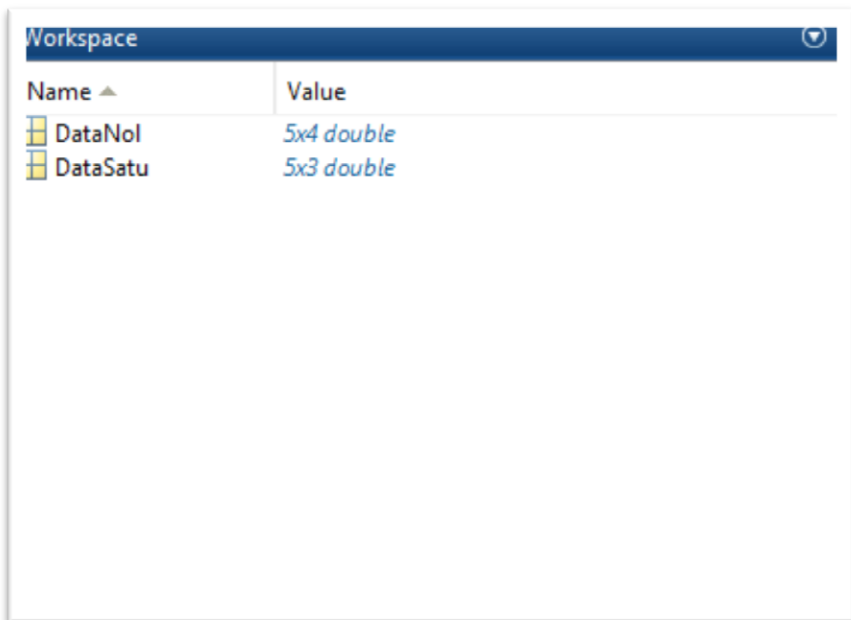
fx >>
```

Gambar 1.2. Command Window Pada Matlab

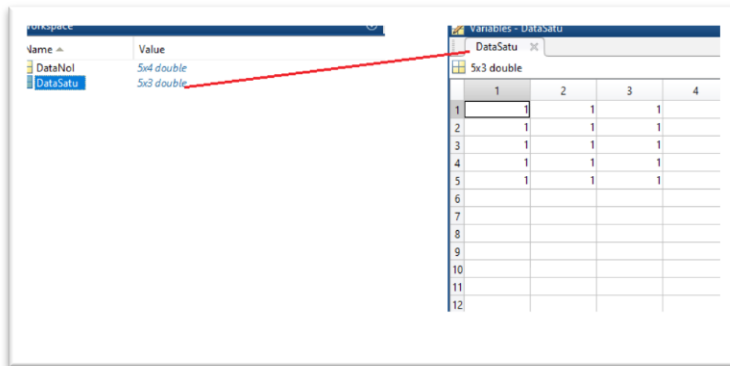
1.3. Workspace pada Matlab

Workspace adalah interface matlab yang memungkinkan kita untuk memperoleh informasi variable apa saja yang tersimpan dalam memori pada saat kita bekerja. Semua variable yang kita gunakan ditampilkan secara detail, mulai dari nama, dimensi, maupun tipe data yang kita gunakan. Workspace juga menampilkan variabel memori yang digunakan pada saat program dijalankan. Artinya kalau kita melakukan debugging program, maka kita dapat melihat perubahan nilai pada setiap variabel memori yang kita gunakan. Gambar 1.3. menunjukkan workspace pada Matlab. Pada workspace terlihat ada dua variabel yang dibentuk melalui command window sebagaimana ditunjukkan pada Gambar 1.2. Jika kita ingin

mengetahui isi dari setiap variabel yang ada pada workspace, kita tinggal click variabel tersebut, maka akan tampil nilai dari variabel sebagaimana ditunjukkan pada Gambar 1.4, karena tidak semua variabel dapat diketahui nilainya pada workspace.



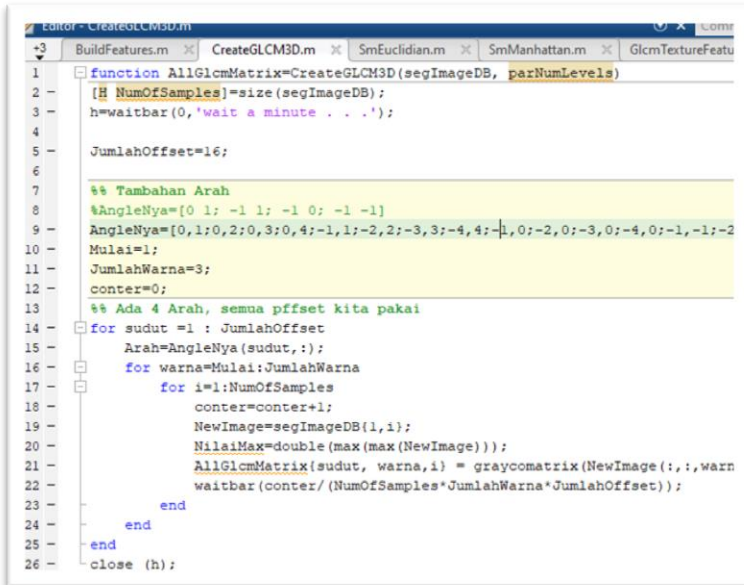
Gambar 1.3. Workspace Pada Matlab



Gambar 1.4. Variabel Pada Workspace

1.4. Editor

Editor merupakan fitur yang memungkinkan kita untuk menuliskan program, menjalankan, menelusuri setiap baris instruksi yang kita tuliskan pada editor, memungkinkan kita membuka beberapa program sekaligus yang disusun berdasarkan tabulasi nama file, sebagaimana ditunjukkan pada Gambar 1.5.

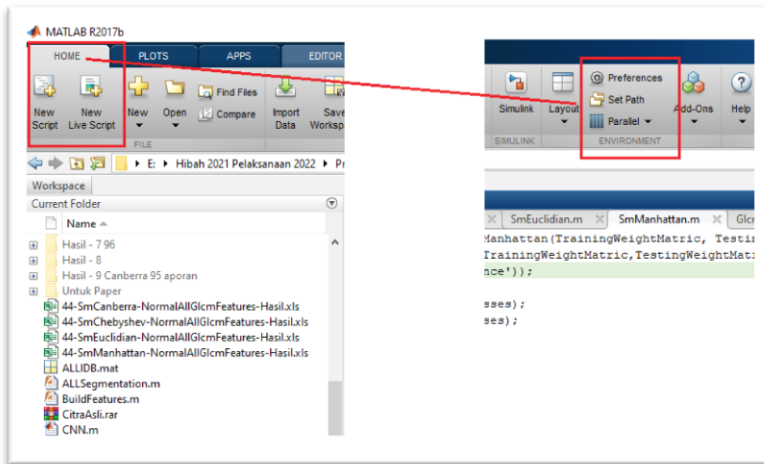


Gambar 1.5. Editor Pada Matlab

1.5. Menu

Tampilan menu yang kami gunakan ini berbeda dengan tampilan Manu generasi sebelumnya. Namun pada prinsipnya mempunyai fitur yang hampir sama. Kami menggunakan Matlab Versi: 9.3.0.713579 (R2017b). Adapun menu yang tersedia pada Matlab ini adalah:

1. Home
2. Plot
3. Apps
4. Editor
5. Publish
6. View

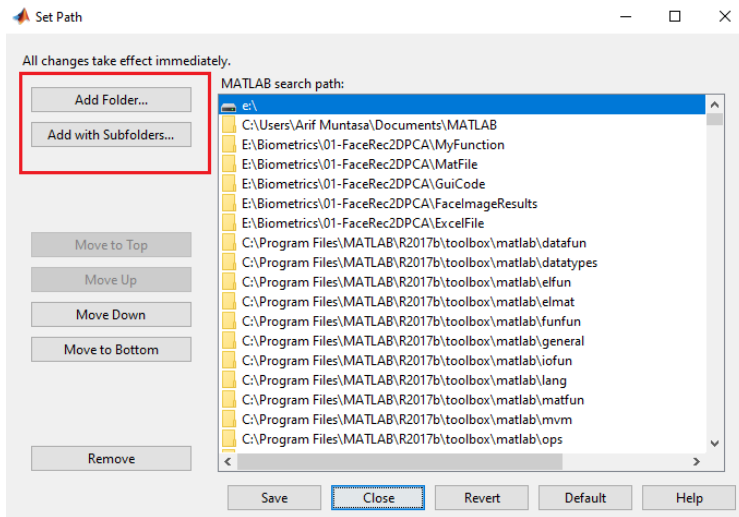


Gambar 1.7. Menu Set Path

Jika kita ingin memasukkan folder sebagai ruang kerja melalui *set path*, maka ada dua hal yang perlu dipahami, diantaranya adalah

1. “*Add Folder*”, fitur ini dimanfaatkan agar semua file yang masuk dalam folder ini dapat dibaca dari lokasi manapun dalam computer yang kita gunakan. Jika kita menggunakan fitur ini, maka file yang dalam subfolder tidak dapat dikenali.
2. “*Add with Subfolders*”, dengan menggunakan fitur ini, maka semua file yang letakkan dalam folder ini, akan dapat dikenali semua termasuk dalam subfolder yang berada dalam folder ini. Fitur yang kedua ini yang sering digunakan oleh para peneliti untuk memeudahkan dalam melakukan eskperimen.

Pada Gambar 1.8 merupakan tampilan *set path* yang telah kami gunakan untuk melakukan eksperimen

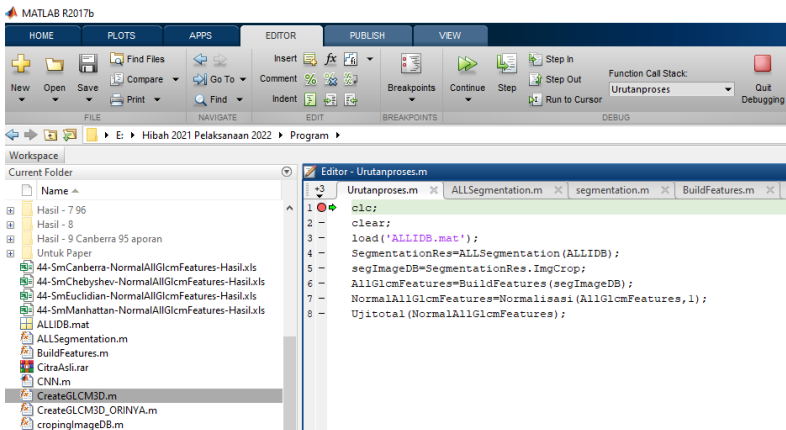


Gambar 1.8. *Set path* Pada Matlab

Pada Gambar 1.8. terlihat ada dua tombol yang dapat digunakan untuk menambahkan folder kerja, agar dapat dikenali dari manapun likasinya kita memanggil.

1.7. Mendebug program menggunakan matlab

Salah satu cara yang paling penting untuk diketahui adalah menelusuri kesalahan program yang kita buat menggunakan fasilitas *debug*. Dengan menggunakan *debug* kita dapat melihat hasil program yang kita buat setiap baris.



Gambar 1.9 *Debug* Program Pada Matlab

Pada Gambar 1.9 terlihat lingkaran merah yang mengindikasikan *debug* dimulai. *Debug* dapat dimulai dari baris manapun, namun jangan sampai *debug* dimulai dari baris yang salah atau sesudahnya, karena kita tidak akan dapat menelusurinya.

- *Step*, tombol dimanfaatkan untuk menjalankan dan menelusuri kode program pada setiap baris. Dengan tombol ini, maka penelusuran akan berjalan dari baris dimana kita meletakkan pertama kali *debug* menuju ke baris selanjutnya
- *Step in*, jika kita ingin mengetahui secara detail *debugging* nya, maka kita bisa menggunakan fitur *step in*, sehingga kita dapat mengetahui secara detail baris demi baris dari fungsi atau library Matlab yang kita gunakan (*built in function*) maupun fungsi yang kita bangun sendiri, sehingga dengan menggunakan *step in* akan membutuhkan waktu yang lebih panjang dibandingkan dengan menggunakan *step*.
- *Step out*, merupakan kebalikan dari *step in*. *Step out* sifatnya mengembalikan *debugging* yang sebelumnya secara detail menjadi normal kembali seperti menggunakan *step*.

- *Run to Cursor*, fasilitas ini bisanya digunakan kalau program yang kita *debug* ada *looping* yang membutuhkan proses lama, maka kita dapat melewati proses *looping* dengan meletakkan kursor dibawah baris akhir looping, sehingga proses looping dijalankan secara normal oleh *Matlab* (bukan dijalankan baris demi baris)

Bab 2. Operator pada Matlab

Pada Matlab, ada 3 operator yang akan dibahas dalam buku ini, diantaranya adalah operator aritmatika, relasional, dan operator logika. Ketiga operator tersebut mempunyai peran yang berbeda. Untuk lebih memahami secara detail, kita akan membahas ketiga operator tersebut.

2.1. Operator aritmatika

Pada kehidupan sehari-hari tentu kita sering mendengar penjumlahan, pengurangan, perkalian, pembagian, dan pemangkatan. Pada matlab kita juga mengenal operasi-operasi tersebut, Berikut ini merupakan beberapa operator dalam Matlab sebagaimana ditunjukkan pada Tabel 2.1.

Tabel 2.1. Operator Pada Matlab

No	Simbol	Keterangan	Contoh
1	+	Penjumlahan	A=10, B=5 A+B=15
2	-	Pengurangan	A=10, B=5 A-B=5
3	*	Perkalian	A=10, B=5 A*B=50
4	/	Pembagian	A=10 B=5 A/B = 2

5	\	Pembagian yang dibalik operannya	$A \setminus B = 0.5$
5	^	Pemangkatan	$A = 10, B = 5$ $A^3 = 1000$ $B^2 = 25$
6	.*	Perkalian antar elemen matrik dengan lokasi sama	$A = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$ $B = [2 \ 1; \ 3 \ 1; \ 5 \ 2]$ $A.*B = [2 \ 2; \ 9 \ 4; \ 25 \ 12]$
7	./	Pembagian antar elemen matrik dengan lokasi sama	$A = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$ $B = [2 \ 1; \ 3 \ 1; \ 5 \ 2]$ $A./B = [0.5 \ 2; \ 1 \ 4; \ 1 \ 3]$
8	.\	Pembagian antar elemen matrik dengan lokasi sama	$A = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$ $B = [2 \ 1; \ 3 \ 1; \ 5 \ 2]$ $A.\setminus B = [2 \ 0.5; \ 1 \ 0.25; \ 1 \ 0.333]$
8	.^	Pemangkatan antar elemen matrik dengan lokasi sama	$A = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$ $A^2 = [1 \ 4; \ 9 \ 16; \ 25 \ 36]$
9	'	Transpose	$A = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$ $A' = [1 \ 3 \ 5; \ 2 \ 4 \ 6]$
10	transpose ()	Transpose	$A = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$ $\text{transpose}(A) = [1 \ 3 \ 5; \ 2 \ 4 \ 6]$

2.1.1 Operator penjumlahan pada matlab

Penjumlahan dapat dilakukan lebih dari dua operan, operator ini berlaku untuk variabel yang bernilai tunggal ataupun matrik. Berikut ini merupakan contoh 3 buah operan dengan 2 operator. Misalkan

$$MatrikA = \begin{pmatrix} 10 & 5 & 3 \\ 4 & 6 & 2 \end{pmatrix}$$

$$MatrikB = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 3 & 8 \end{pmatrix}$$

$$MatrikA = \begin{pmatrix} 8 & 3 & 2 \\ 9 & 6 & 1 \end{pmatrix}$$

Maka hasil penjumlahan ke 3 matrik tersebut adalah

>> MatrikA=[10 5 3; 4 6 2]

MatrikA =

10 5 3

4 6 2

>> MatrikB=[1 4 7; 2 3 8]

MatrikB =

1 4 7

2 3 8

```
>> MatrikC=[8 3 2; 9 6 1]
```

MatrikC =

```
8 3 2
```

```
9 6 1
```

```
>> Hasil=MatrikA+MatrikB+MatrikC
```

Hasil =

```
19 12 12
```

```
15 15 11
```

Matlab juga dapat menyelesaikan penjumlahan nilai tunggal dengan matrik, misalkan **MatrikA** kita jumlahkan dengan suatu nilai 10, maka itu artinya sama dengan nilai tiap elemen atriik ditambahkan dengan 10, maka hasilnya adalah

```
>> Hasil=MatrikA+10
```

Hasil =

```
20 15 13
```

```
14 16 12
```

2.1.2 Operator pengurangan pada Matlab

Serupa dengan penjumlahan, pengurangan pada matlab juga dapat dilakukan dengan menggunakan beberapa operan. Kita akan menggunakan contoh yang sama dengan Penjumlahan matriknya. Misalkan

>> Hasil = MatrikA - MatrikB

Hasil =

9 1 -4

2 3 -6

>> Hasil = MatrikA - MatrikB - MatrikC

Hasil =

1 -2 -6

-7 -3 -7

Pada operasi pengurangan, matrik juga dapat dikurangkan dengan nilai tunggal ataupun sebaliknya, misalkan **MatrikA** kita kurangkan dengan 10, maka hasilnya adalah

>> Hasil = MatrikA - 10

Hasil =

0 -5 -7

-6 -4 -8

2.1.3 Operator perkalian dan perkalian antar elemen pada matlab

Syarat perkalian antar matrik adalah kolom dari matrik yang pertama harus sama dengan baris pada matrik kedua. Jika syarat ini tidak dapat dipenuhi, maka operasi perkalian pasti error, karena dimensi yang dipersyaratkan tidak sesuai. Misalkan perkalian antara *MatrikA* dan *MatrikB* berikut ini


```
>> MatrikA
```

```
MatrikA =
```

```
10  5  3
```

```
4   6  2
```

```
>> MatrikB
```

```
MatrikB =
```

```
1   4  7
```

```
2   3  8
```

```
>> MatrikC
```

```
MatrikC =
```

```
8   3  2
```

```
9   6  1
```

```
>> Hasil=MatrikA*MatrikC
```

Error using *

Inner matrix dimensions must agree.

Error ini disebabkan karena dimensi dari kolom *MatrikA* sebagai operan pertama tidak sama dengan baris dari *MatrikC* sebagai operan kedua, Sekarang kita transpose *MatrikC*, selanjutnya kita kalikan Kembali, maka hasilnya

```
>> TransMatrikC=MatrikC'
```

```
TransMatrikC =
```

```
8 9
```

```
3 6
```

```
2 1
```

```
>> Hasil=MatrikA*TransMatrikC
```

```
Hasil =
```

```
101 123
```

```
54 74
```

Matlab juga menyediakan perkalian antar elemen yang bersesuaian dengan posisi masing-masing matrik dari operan satu dan dua, sehingga syarat perkalian antar elemen adalah mempunyai dimensi sama, baik untuk baris dan kolom. Misalkan perkalian antar elemen antara *MatrikA* dan *MatrikB*, maka dapat dituliskan menggunakan perintah '*.**'. Perintah perkalian, namun didepannya didahului dengan tanda dot ('.'), sebagaimana contoh dalam Matlab berikut ini:

```
>> MatrikA
```

```
MatrikA =
```

```
10 5 3
```

```
4 6 2
```

>> **MatrikB**

MatrikB =

1 4 7

2 3 8

>> **Hasil=MatrikA.*MatrikB**

Hasil =

10 20 21

8 18 16

2.1.4 operator pembagian pada Matlab

Pembagian antara bilangan tunggal dengan bilangan tunggal tidak menjadi masalah. Bagaimana kalau matrik dibagi dengan matrik. Pembagian matrik dengan matrik sama dengan mengalikan matrikan dengan inversnya.

$$A = \frac{B}{C} = B \times C^{-1} \quad (2.1)$$

Pada Persamaan (2.1) terlihat bahwa suatu matrik B dibagi dengan matrik c sama dengan perkalian antara matrik B dengan invers dari matrik C.

Misalkan dua buah matrik B dan C sebagaimana pada matrik dibawah ini

$$B = \begin{pmatrix} 1 & 4 & 2 \\ 3 & 6 & 3 \\ 5 & 4 & 4 \end{pmatrix}$$

$$C = \begin{pmatrix} 3 & 2 & 12 \\ 2 & 7 & 10 \\ 9 & 9 & 5 \end{pmatrix}$$

Hasil pembagian antara matrik B dengan C adalah

```
>> B=[1 4 2; 3 6 3; 5 4 4]
```

B =

1 4 2

3 6 3

5 4 4

```
>> C=[3 2 12; 2 7 10; 9 9 5]
```

C =

3 2 12

2 7 10

9 9 5

```
>> A=B*inv(C)
```

A =

-0.3211 0.5358 0.0991

-0.3303 0.5339 0.3248

0.2477 -0.1505 0.5064

Matlab juga mengenali operasi pembagian matrik menggunakan karakter '/', sebagaimana hasil berikut ini

>> $A=B/C$

$A =$

-0.3211 0.5358 0.0991

-0.3303 0.5339 0.3248

0.2477 -0.1505 0.5064

2.2. Operator relasional

Operator relasional digunakan untuk membandingkan antara satu nilai dengan lainnya. Operator digunakan dalam penyeleksian kondisi.

Tabel 2.1. Operator Relasional

No	Operator Relasional	Keterangan
1	==	Sama dengan
2	~=	Tidak sama dengan
3	>	Lebih besar
4	<	Lebih kecil
5	>=	Lebih besar sama dengan
6	<=	Lebih kecil sama dengan

2.3. Operator logical

Operator yang terakhir adalah operator logika, dimana operator ini juga dipakai dalam pemrograman ketika penyeleksian kondisi atau pengambilan keputusan. Adapun

Tabel 2.3. Operator Logika

No	Operator Logika	Makna
1	&&	And
2		Or
3	~	Not

Sebelum kita mencoba menggunakan perintah Matlab, kami akan memberikan penjelasan dulu untuk masing-masing-operator menggunakan table, sebagaimana terlihat pada Tabel 2.4 Nilai 1 merepresentasikan kondisi *true*, sebaliknya kondisi *false* direpresentasikan dengan nilai 0.

Tabel 2.3. Operator Logika

No	A	~A	B	~B	A && B	A B
1	1	0	1	0	1	1
2	1	0	0	1	0	1
3	0	1	1	0	0	1
4	0	1	0	1	0	0

Kita mencoba menggunakan matlab untuk ketiga operator logika tersebut sebagaimana terlihat pada berikut ini

>> Variabel1

Variabel1 =

100