

Pengenalan GNU Octave

Perangkat Lunak Gratis untuk Komputasi Numerik dan Visualisasi Data

Saifuddin Arief

Lisensi Dokumen:

Copyright © 2003-2008 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

1. Sekilas Mengenai Octave

GNU Octave adalah suatu perangkat lunak gratis (*freeware*) dan bahasa tingkat tinggi untuk komputasi numerik dan visualisasi data. Octave dirancang sebagai tiruan dari Matlab. Sumber informasi mengenai Octave dapat dilihat pada website www.octave.org.

Pada awalnya Octave dikembangkan oleh John W. Eaton (Universitas Texas) dan sekarang pengembangan dan pemeliharaan Octave dilakukan oleh beberapa orang *volunteer* dari berbagai penjuru dunia.

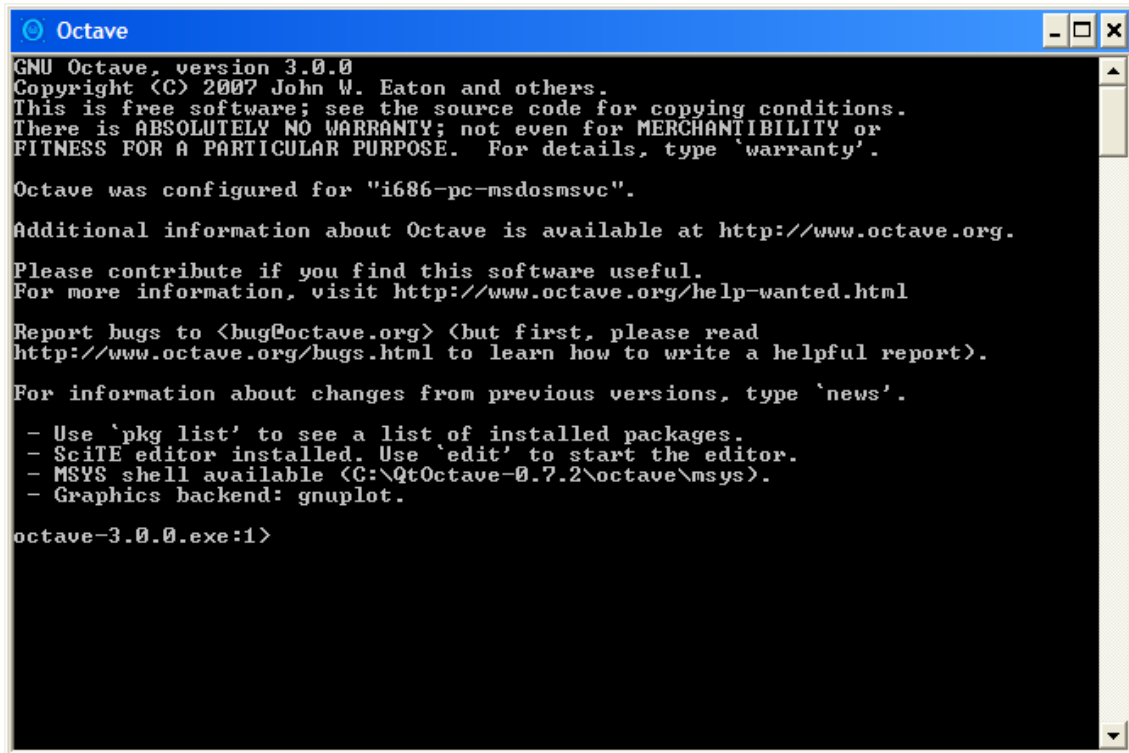
Kelebihan utama dari Octave yaitu gratis (*freeware*) dan tersedia untuk berbagai sistem operasi seperti Windows 98/2000/XP, Mac OS/X, Debian, Suse, Fedora, RedHat Linux.

2. Menjalankan program Octave

Pada kebanyakan sistem operasi program GNU Octave dapat dijalankan dengan memberikan perintah octave pada *shell command*. Setelah perintah tersebut kita berikan maka akan muncul suatu jendela GNU Octave. Pada jendela tersebut akan ditampilkan beberapa pesan singkat mengenai Octave dan kemudian di bawah pesan singkat tersebut

ditampilkan sebuah *prompt*, yang menandakan bahwa Octave siap untuk menerima perintah yang akan kita berikan.

Gambar 1 adalah tampilan dari program Octave untuk sistem operasi Windows.



```
GNU Octave, version 3.0.0
Copyright (C) 2007 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "i686-pc-msdosmsvc".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type 'news'.

- Use 'pkg list' to see a list of installed packages.
- SciTE editor installed. Use 'edit' to start the editor.
- MSYS shell available (C:\QtOctave-0.7.2\octave\msys).
- Graphics backend: gnuplot.

octave-3.0.0.exe:1>
```

Gambar 1. Tampilan Jendela Octave pada Sistem Operasi Windows

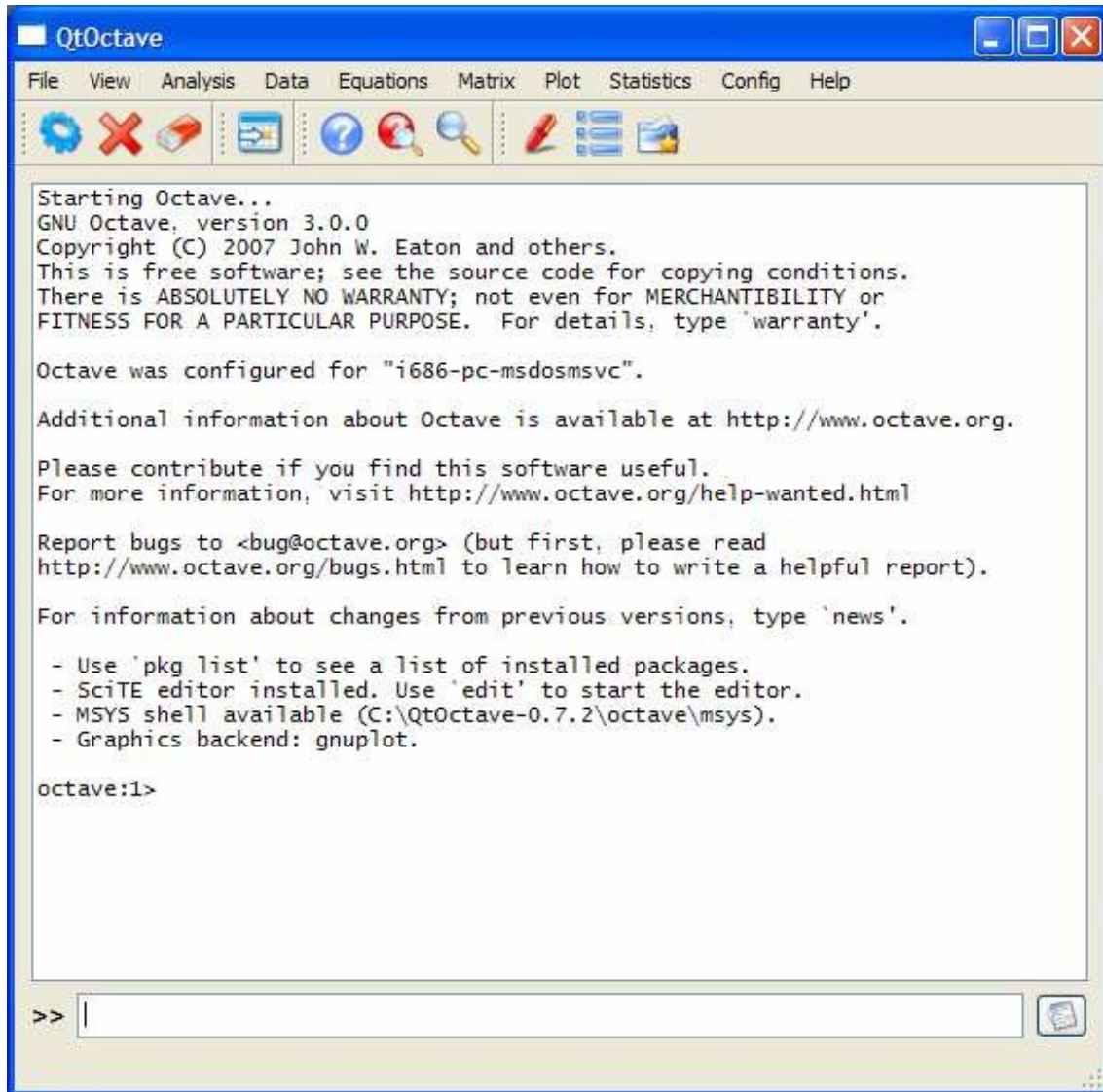
Untuk keluar dari program Octave gunakan perintah quit atau exit.

3. QtOctave

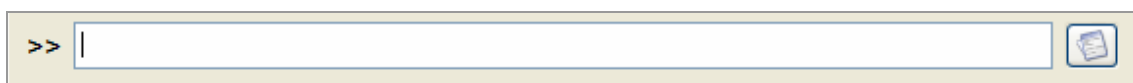
QtOctave adalah merupakan sebuah antara muka grafis yang dikembangkan untuk program Octave. Antara muka grafis ini dikembangkan untuk menambahkan beberapa fasilitas yang tidak terdapat pada program Octave yang langsung dijalankan dari *shell command* sehingga program Octave lebih mudah digunakan. Gambar 2 merupakan tampilan dari QtOctave.

Pada QtOctave, perintah-perintah yang kita berikan tidak dimasukkan secara langsung pada baris perintah, melainkan pada kotak teks masukkan yang terdapat pada bagian bawah dari jendela QtOctave. (Lihat Gambar 3)

Untuk keluar dari program QtOctave gunakan menu File Exit .



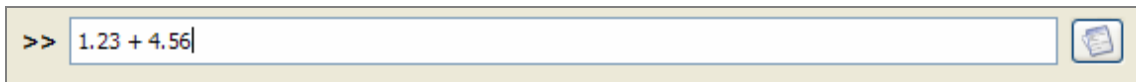
Gambar 2. Jendela QtOctave



Gambar 3. Kotak Teks Masukan

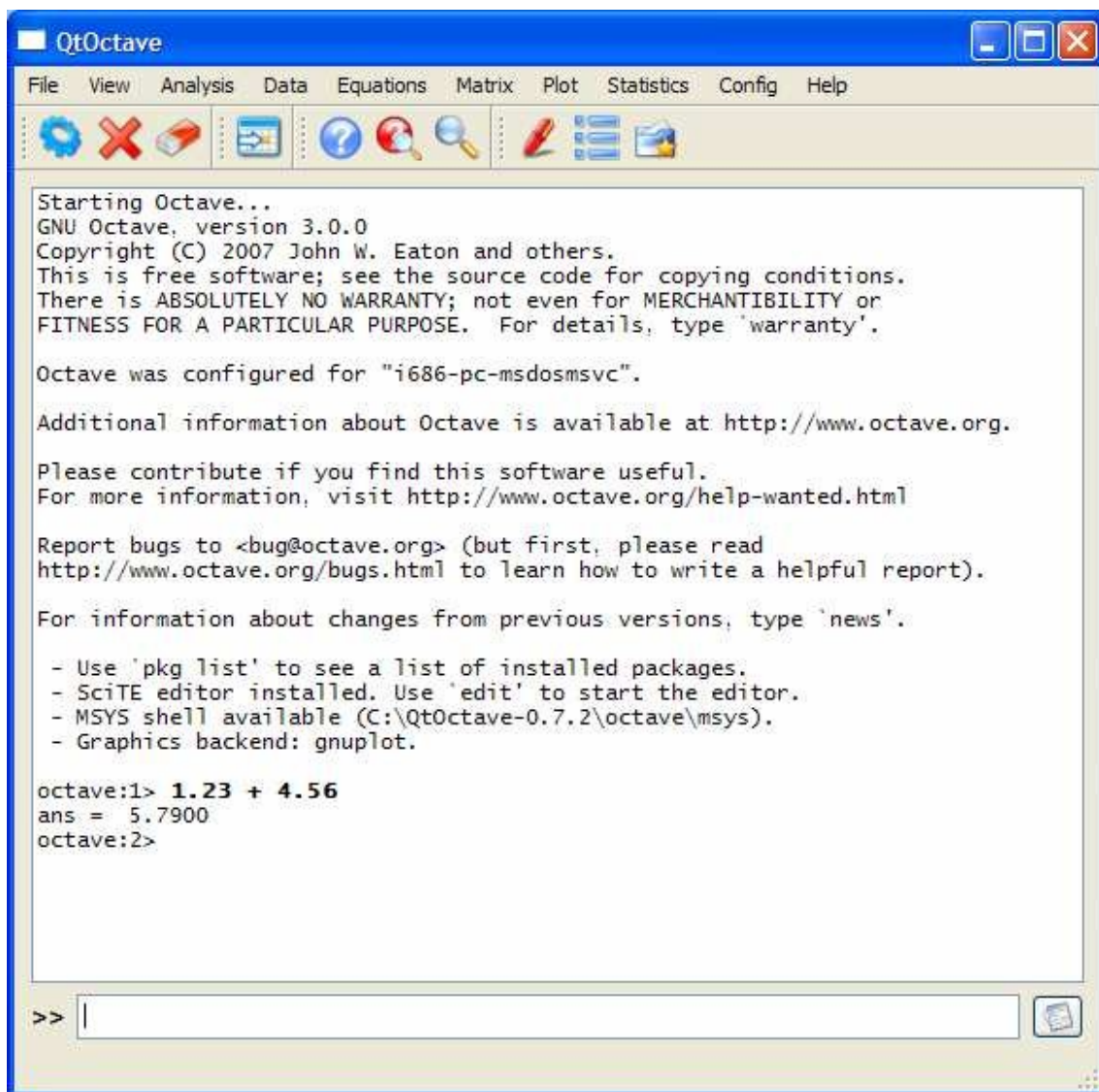
4. Contoh sebuah perintah sederhana

Misalkan kita akan melakukan suatu perhitungan yaitu $1.23 + 4.56$ maka kita harus menuliskan ekspresi matematika tersebut pada Kotak teks masukan kemudian tekan tombol **ENTER** untuk melakukan eksekusi terhadap ekspresi matematika yang telah kita ketikkan.



Gambar 4. Contoh sebuah perintah sederhana

Perintah yang telah kita masukkan dan hasilnya akan ditampilkan pada jendela utama, seperti yang ditunjukkan pada gambar di bawah ini.



Gambar 5.

5. Dokumentasi dan Sistem Bantuan

Octave juga dilengkapi dengan dokumentasi yang menjelaskan program Octave dengan cukup detail.

Selain dari dokumentasi, untuk mendapat penjelasan mengenai suatu perintah atau juga dapat dilakukan dengan menggunakan perintah `help` kata-kunci, dimana kata-kunci adalah nama fungsi atau operator yang akan kita cari penjelasannya. Berikut ini adalah contoh dari penggunaan perintah `help`.

```
octave:2> help inv
-- Loadable Function: [X, RCOND] = inv (A)
-- Loadable Function: [X, RCOND] = inverse (A)
Compute the inverse of the square matrix A. Return an estimate
of the reciprocal condition number if requested, otherwise warn
of an ill-conditioned matrix if the reciprocal condition number
is small.
Overloaded function:
spinv (sparse matrix, ...)
spinv (sparse complex matrix, ...)
spinv (sparse bool matrix, ...)
ginv (galois, ...)
ginv (galois, ...)
spinv (sparse matrix, ...)
spinv (sparse complex matrix, ...)
spinv (sparse bool matrix, ...)
inv is a built-in function
Additional help for built-in functions and operators is
available in the on-line version of the manual. Use the command
`doc <topic>` to search the manual index.
Help and information about Octave is also available on the WWW
at http://www.octave.org and via the help@octave.org
mailing list.

octave:3> help *
*** *.
Multiplication operator. See also `.*`
Additional help for built-in functions and operators is
available in the on-line version of the manual. Use the command
`doc <topic>` to search the manual index.
Help and information about Octave is also available on the WWW
at http://www.octave.org and via the help@octave.org
mailing list.
```

6. Variabel dan Ekspresi

Variabel adalah sebuah nama yang digunakan untuk menyimpan nilai suatu obyek. Notasi statemen penugasan adalah sebagai berikut:

`x = ekspresi`

dimana `x` adalah nama variabel dan ekspresi adalah suatu ekspresi matematika. Nama variabel di dalam Octave adalah bersifat sensitif terhadap ukuran huruf, sehingga `xawal` dan `Xawal` adalah dua buah variabel yang berbeda.

Selain variabel-variabel yang dapat kita buat sendiri, di dalam Octave telah terpasang beberapa variabel khusus yang menyatakan suatu konstanta matematika, seperti `pi` untuk $\pi = 3.1415927\dots$, `i` atau `j` untuk $i = \sqrt{-1}$ serta `e` untuk $e = 2.7182818\dots$

Nilai dari suatu ekspresi yang kita masukkan akan ditampilkan pada baris berikutnya, kecuali jika pada akhir ekspresi tersebut kita tambahkan tanda titik koma (;).

Apabila kita mempunyai suatu ekspresi yang cukup panjang dan tidak cukup untuk ditulis pada satu baris, maka kita harus menggunakan tanda titik tiga (...) pada akhir ekspresi, sebagai tanda bahwa ekspresi bersambung pada baris berikutnya.

Untuk memperjelas perintah yang kita masukkan kita dapat menambahkan suatu baris komentar. Baris komentar dibuat dengan menggunakan tanda % atau #. Segala sesuatu di belakang tanda tersebut akan diabaikan oleh Octave. Baris komentar dapat ditulis pada suatu baris tersendiri atau di belakang suatu ekspresi.

```
octave:1> lebar = 12.5
lebar = 12.500

octave:2> tinggi = 8;

octave:3> luas = lebar * tinggi
luas = 100

octave:4> r = 10; % diameter

octave:5> A = pi*r^2 # luas lingkaran
A = 314.16

octave:6> % Contoh penggunaan simbol tiga titik (...)
octave:6> S = 1 + 1/2 + 1/3 + 1/4 + 1/5 + ...
1/6 + 1/7 + 1/8 + 1/9 + 1/10
S = 2.9290
```

7. Ruang Kerja

Variabel-variabel yang telah kita buat akan disimpan oleh Octave dalam ruang kerja. Untuk melihat nama-nama variabel yang telah dibuat, gunakan perintah who

```
octave:7> who
*** dynamically linked functions:
  __COM__ builtin:find dispatch getpwuid
*** currently compiled functions:
  __default_graphics__ index rindex
  edit ispc strcat
  fileparts isunix strrep
  findstr lower
  fullfile pkg
*** local user variables:
  A _margin_ lebar r
  S home_path luas tinggi
```

Terlihat bahwa perintah who, selain menampilkan variabel-variabel yang telah kita buat, juga menampilkan variabel-variabel yang telah terpasang pada Octave.

Untuk menghapus suatu variabel, gunakan perintah clear .

```
octave:8> clear luas tinggi      % menghapus variabel luas dan tinggi
octave:9> clear                  % menghapus semua variabel
```

8. Operator-operator dan Fungsi-fungsi Matematika

Operator-operator untuk perhitungan aritmatika adalah sama dengan operator-operator yang terdapat pada kalkulator atau perangkat lunak lainnya yaitu $+$, $-$, $*$, $/$ dan $^$. Dimana tanda-tanda tersebut masing-masing adalah simbol untuk operasi penjumlahan, pengurangan, perkalian, pembagian serta pemangkatan. Di dalam Octave juga telah terpasang sejumlah fungsi-fungsi yang diperlukan dalam perhitungan matematika, seperti `sqrt`, `abs`, `exp`, `sin`, `cos`, `tan` dan lain sebagainya.

Berikut ini adalah contoh-contoh perhitungan matematika.

```
octave:10> (1 + sqrt(5))/2
ans = 1.6180

octave:11> Tc = 372.7*(1 + 1/(1.242 + 1.067))
Tc = 534.11

octave:12> dHv = (7.08*(1 - 0.6939)^0.354 + ...
10.95*0.2559*(1 - 0.6939)^0.456)*0.008314*504.4
dHv = 26.375

octave:13> V = 0.773*sqrt(1.4*8314*261.6/29)
V = 250.48

octave:14> dx = 109*cos(35/180*pi)
dx = 89.288

octave:15> TB = 310.9*(log(4.506e6) + (1 - 1.434)*log(1/5528))/log(4.506e6)
TB = 386.79
```

9. Bilangan Kompleks

Octave juga dapat menangani bilangan kompleks dan operasi-operasi matematikanya. Bilangan kompleks dinyatakan dengan notasi $a + b*i$, dimana a adalah komponen real dan b adalah komponen imajineranya.

```
octave:16> z1 = 6 - 8*i
z1 = 6 - 8i

octave:17> z2 = 3 + i
z2 = 3 + 1i

octave:18> p = z1 + z2
p = 9 - 7i

octave:19> q = z1 - z2
q = 3 - 9i

octave:20> abs(z1)
ans = 10

octave:21> z1 * z2
ans = 26 - 18i

octave:22> z2 / z1
ans = 0.10000 + 0.30000i
```

10. Matrik dan Vektor

Salah satu kelebihan Octave yaitu kemampuannya dalam menangani berbagai macam operasi manipulasi terhadap data yang berupa suatu matrik. Pada dasarnya semua data numerik di dalam Octave dianggap sebagai suatu matrik. Vektor dan skalar merupakan bentuk khusus dari suatu matrik. Vektor adalah suatu matrik yang hanya mempunyai satu baris atau satu kolom saja, sementara itu skalar adalah suatu matrik yang hanya terdiri dari satu elemen saja.

Pembuatan data matrik dan vektor secara manual dilakukan dengan menggunakan operator kurung siku (`[...]`). Dimana elemen-elemen matrik atau vektor dimasukkan diantara kedua kurung siku tersebut. Untuk memisahkan elemen yang satu dengan elemen yang lainnya yang terletak pada satu baris dapat digunakan tanda koma (`,`) atau tanda spasi. Kemudian untuk memisahkan antara baris yang satu dengan yang lainnya, gunakan tanda titik koma (`;`) atau tanda ENTER

Contoh-contoh pembuatan matrik dan vektor adalah sebagai berikut:

```
octave:23> X = [1 2 3; 4 5 6; 7 8 9]
X =
  1 2 3
  4 5 6
  7 8 9
```

```
octave:24> X = [1,2,3          % Cara yang lain
4,5,6
7,8,9]
X =
  1 2 3
  4 5 6
  7 8 9
```

```
octave:25> v = [1 2 3 4 5] % vektor baris
v =
  1 2 3 4 5
```

```
octave:26> w = [3;4;1;2] % vektor kolom
w =
  3
  4
  1
  2
```

Untuk membuat suatu vektor baris dimana nilai dari elemen-elemennya berubah secara konstan dari suatu awal nilai tertentu sampai nilai akhir tertentu kita dapat menggunakan operator tanda titik dua (`:`). Notasi pembuatan vektor indek adalah `i:j:k`. Apabila nilai `j` sama maka notasi tersebut dapat ditulis dengan notasi yang lebih singkat yaitu `i:k`.

```
octave:27> ii = 1:10
ii =
  1 2 3 4 5 6 7 8 9 10
```

```
octave:28> m = 0:0.25:1
m =
  0.00000 0.25000 0.50000 0.75000 1.00000
```

```
octave:29> mm = 1:2:10
mm =
  1 3 5 7 9
```


Octave juga menyediakan sejumlah fungsi yang dapat digunakan untuk membuat matrik-matrik khusus. Ilustrasi dari fungsi-fungsi tersebut diberikan pada contoh-contoh di bawah ini.

```
octave:30> A = zeros(3,4)    % matrik nol
A =
0 0 0 0
0 0 0 0
0 0 0 0

octave:31> B = ones(2,5)    % matrik satuan
B =
1 1 1 1 1
1 1 1 1 1

octave:32> Y = eye(3,3)      % matrik identitas
Y =
1 0 0
0 1 0
0 0 1

octave:33> D = diag(1:4)     % matrik diagonal
D =
1 0 0 0
0 2 0 0
0 0 3 0
0 0 0 4

octave:34> R = rand(5,7)     % matrik bilangan random
R =
Columns 1 through 6:
0.8032295 0.3968942 0.0382533 0.2302370 0.8103967 0.4180864
0.5306356 0.6840040 0.0021653 0.3550691 0.8546796 0.0852024
0.0060400 0.8822578 0.3772114 0.3725500 0.6503088 0.0069846
0.1495583 0.6815509 0.5379771 0.5032904 0.3119187 0.1095100
0.6878194 0.7801776 0.6381595 0.2810518 0.1608820 0.3831782
Column 7:
0.8407592
0.6283008
0.6465942
0.9465550
0.9109141
```

11. Operasi Berbasis Vektor

Di dalam Octave, secara umum operasi-operasi matematika terhadap obyek matrik dan vektor dapat dilakukan dengan sangat mudah tanpa harus menggunakan suatu perulangan, sebagaimana yang diilustrasikan pada contoh-contoh di bawah ini.

```
octave:35> x = 0:pi/4:pi
x =
0.00000 0.78540 1.57080 2.35619 3.14159

octave:36> cos(x)
ans =
1.0000e+000 7.0711e-001 6.1232e-017 -7.0711e-001 -1.0000e+000

octave:37> u = [1 2 3];

octave:38> y = exp(u)
y =
2.7183 7.3891 20.0855

octave:39> z = log(y)
z =
1 2 3
```

```
octave:40> w = round(y)
w =
3 7 20
```

Operasi Aljabar Linier, seperti penjumlahan, pengurangan dan perkalian, juga dapat dilakukan dengan sangat mudah tanpa harus menggunakan suatu ekspresi perulangan secara eksplisit. Pada operasi aljabar argumen-argumennya harus mempunyai dimensi yang kompatibel, jika dimensi tidak kompatibel maka operasinya tidak dapat dieksekusi dan Octave akan menampilkan suatu pesan kesalahan.

```
octave:41> X = [9 8 5; 1 3 0; 2 4 6]
X =
9 8 5
1 3 0
2 4 6
```

```
octave:42> Y = [3 2 1; 4 5 6; 9 8 7]
Y =
3 2 1
4 5 6
9 8 7
```

```
octave:43> A = X + Y
A =
12 10 6
5 8 6
11 12 13
```

```
octave:44> B = X - Y
B =
6 6 4
-3 -2 -6
-7 -4 -1
```

```
octave:45> p = [1 2; 3 8]
p =
1 2
3 8
```

```
octave:46> Z = A + p
octave:46>error: operator +: nonconformant arguments
(op1 is 3x3, op2 is 2x2)
error: evaluating binary operator `+' near line 46, column 7
error: evaluating assignment expression near line 46, column 3
```

```
octave:46> i = eye(2,2)
i =
1 0
0 1
```

```
octave:47> j = [6;4]
j =
6
4
```

```
octave:48> k = ones(3,3)
k =
1 1 1
1 1 1
1 1 1
```

```
octave:49> f = p*i
f =
1 2
3 8
```

```
octave:50> g = p*j
g =
14
50
```

```
octave:51> ln ==ppkk
octave:51>error: operator *: nonconformant arguments
(op1 is 2x2, op2 is 3x3)
error: evaluating binary operator `*' near line 51, column 6
error: evaluating assignment expression near line 51, column 3
```

Selain operasi perkalian yang mengikuti aturan dalam Aljabar Linier, kita juga dapat melakukan operasi terhadap obyek matrik dan vektor dengan melakukan operasi secara elemen dengan elemen. Operasi elemen dengan elemen juga dapat diterapkan pada operasi pembagian. Notasi untuk perkalian dan pembagian secara elemen dengan elemen adalah .* dan ./ .

```
octave:51> x = [1 2; 3 4]
x =
  1 2
  3 4

octave:52> y = [5 6; 7 8]
y =
  5 6
  7 8

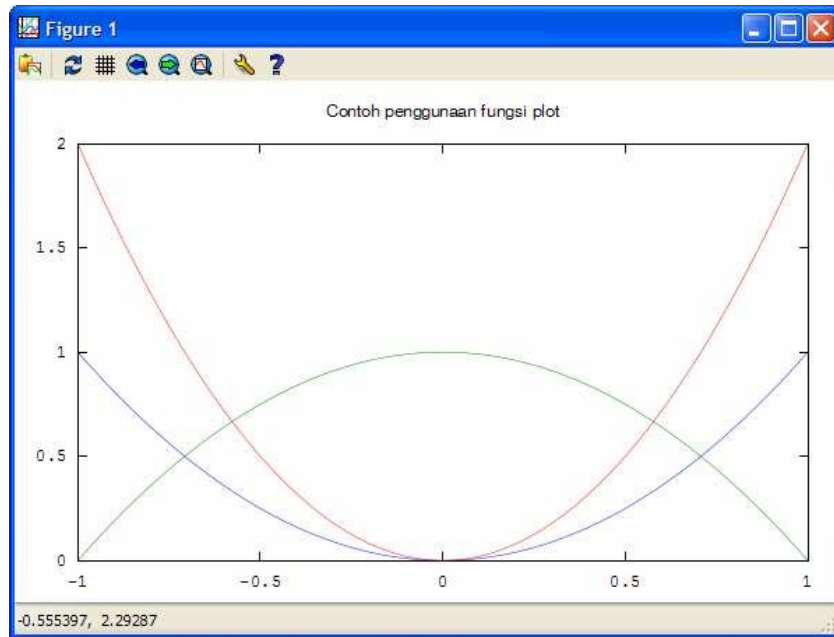
octave:53> u = x.*y
u =
  5 12
  21 32

octave:54> z = y./x
z =
  5.0000 3.0000
  2.3333 2.0000
```

12. Visualisasi Data

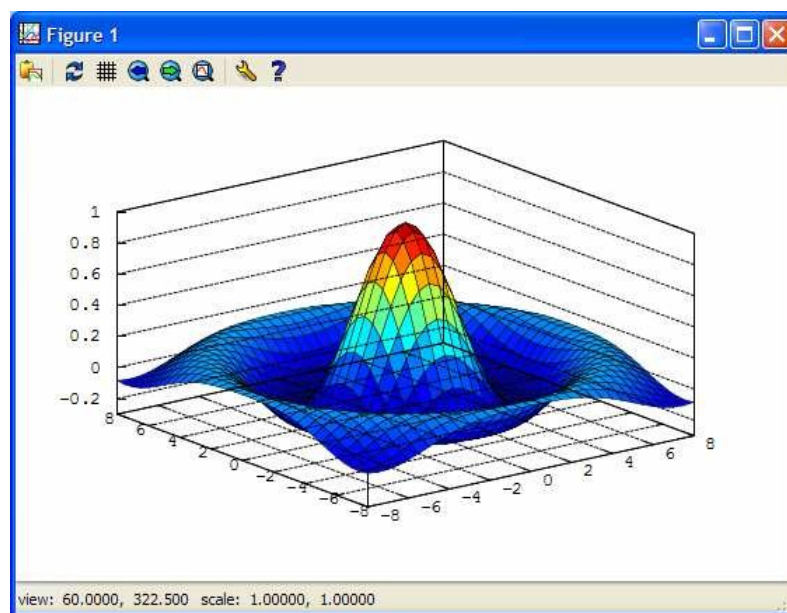
Octave dapat digunakan untuk melakukan visualisasi data, baik secara dua dimensi maupun tiga dimensi. Untuk membuat grafik dua dimensi kita dapat menggunakan perintah plot, kemudian untuk grafik tiga dimensi kita dapat menggunakan perintah surf, mesh, seperti yang diperlihatkan pada contoh-contoh di bawah ini. Grafik yang dihasilkan oleh perintah-perintah grafis akan ditampilkan pada jendela tersendiri, yaitu jendela grafik. Penjelasan detail mengenai pembuatan grafik dapat dilihat pada sistem bantuan yang terdapat pada program Octave.

```
octave:55> % Contoh penggunaan fungsi plot (lihat Gambar 6)
octave:55> x = linspace(-1,1,61)';
octave:56> y1 = x.^2; y2 = 1 - y1; y3 = 2*y1;
octave:57> plot(x,[y1 y2 y3])
octave:58>title('Contoh penggunaan fungsi plot')
```



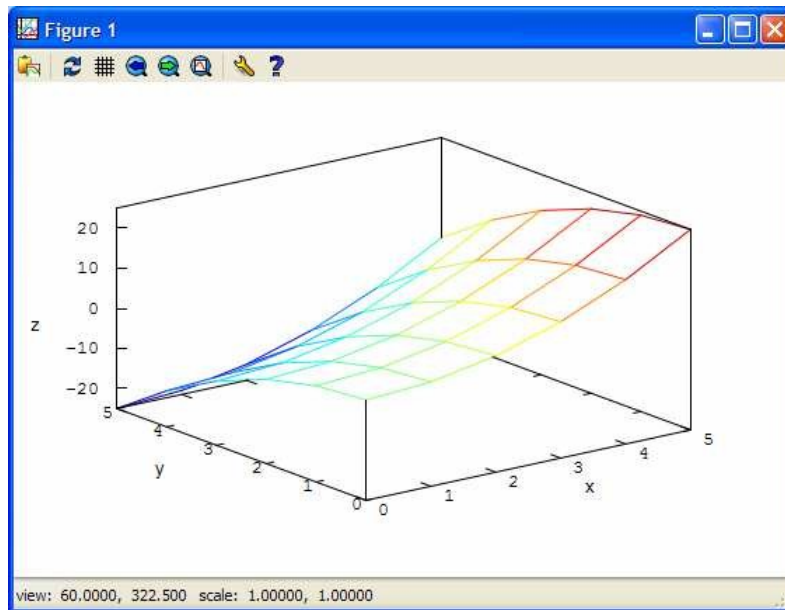
Gambar 6

```
octave:62> % Contoh penggunaan fungsi surf (lihat Gambar 7)
octave:62> [x y] = meshgrid(-8:0.5:8);
octave:63> r = sqrt(x.^2 + y.^2) + eps;
octave:64> z = sin(r)./r;
octave:65> surf(x,y,z)
```



Gambar 7

```
octave:66> % Contoh penggunaan fungsi mesh (lihat Gambar 8)
octave:66> [x y] = meshgrid(0:5);
octave:67> z = x.^2 - y.^2;
octave:68> mesh(x,y,z)
octave:69> xlabel('x'), ylabel('y'), zlabel('z')
```



Gambar 8

10. Komputasi Numerik

Penyelesaian berbagai macam persoalan dalam komputasi numerik dapat dilakukan dengan mudah, seperti yang ditunjukkan pada contoh-contoh di bawah ini.

Sistem persamaan linear $Ax = b$ dapat diselesaikan dengan mudah menggunakan operator pembagian kiri (\backslash). Notasi $A \backslash b$ adalah ekuivalen dengan $inv(A)*b$.

```
octave:70> A = [1 2 1 4; 2 0 4 3; 4 2 2 1; -3 1 3 2]
A =
  1 2 1 4
  2 0 4 3
  4 2 2 1
 -3 1 3 2
```

```
octave:71> b = [13; 28; 20; 6]
b =
 13
 28
 20
  6
```

```
octave:72> x = A\b
x =
  3.0000
 -1.0000
  4.0000
  2.0000
```

Nilai determinan dan rank dari matrik A dapat dihitung dengan menggunakan fungsi `det(A)` dan `rank(A)` .

```
octave:73> d = det(A)
d = -180.00

octave:74> n = rank(A)
n = 4
```

Inversi dari suatu matrik A dapat kita hitung dengan menggunakan fungsi `inv(A)` .

```
octave:75> B = inv(A)
B =
-7.3997e-018  8.3333e-002  8.3333e-002 -1.6667e-001
 6.6667e-002 -3.4444e-001  3.2222e-001  2.2222e-001
-2.0000e-001  1.1667e-001  1.1667e-001  1.6667e-001
 2.6667e-001  1.2222e-001 -2.1111e-001 -1.1111e-001

octave:76> A*B
ans =
 1.00000 -0.00000 -0.00000 -0.00000
-0.00000  1.00000 -0.00000  0.00000
-0.00000  0.00000  1.00000 -0.00000
 0.00000 -0.00000 -0.00000  1.00000
```

Untuk mencari penyelesaian persamaan nonlinier $f(x) = 0$, kita dapat menggunakan fungsi `fsolve` , seperti yang digambarkan pada contoh di bawah ini.

```
octave:79> function y = f(x)
  y = x - exp(-x);
endfunction

octave:82> [x0,f0,info] = fsolve("f",2)
x0 = 0.56714
f0 = 1.1102e-016
info = 1
```

Integrasi numerik dari suatu fungsi $f(x)$ dapat dilakukan dengan menggunakan fungsi `quad`, seperti yang diperlihatkan pada contoh di bawah ini.

```
octave:1> function y = f(x)
> y = x^3/(exp(x) - 1);
> endfunction

octave:2> [v,ierr,nfun,err] = quad("f",0,5)
v = 4.8999
ierr = 0
nfun = 21
err = 5.4400e-14

octave:3> function y = f(x)
> y = exp(-x^2/2)/(sqrt(2*pi));
> endfunction

octave:4> [v,ierr,nfun,err] = quad("f",-inf,0)
v = 0.50000
ierr = 0
nfun = 105
err = 5.0891e-09
```

11. Pemrograman

Octave menyediakan sejumlah kontrol pemrograman yang dapat kita gunakan untuk mengatur jalannya eksekusi suatu program dengan menggunakan statemen perulangan dan kondisional. Pada umumnya statemen perulangan dan kondisional digunakan dalam sebuah skrip atau fungsi.

Ilustrasi penggunaan statemen perulangan for dan kondisional if diberikan pada contoh-contoh di bawah ini.

```
octave:1> # Contoh statemen perulangan for
```

```
octave:1> H = zeros(4,4);
```

```
octave:2> for i = 1:4
    for j = 1:4
        H(i,j) = 1/(i+j-1);
    end
end
```

```
octave:3> H
H =
1.00000 0.50000 0.33333 0.25000
0.50000 0.33333 0.25000 0.20000
0.33333 0.25000 0.20000 0.16667
0.25000 0.20000 0.16667 0.14286
```

```
octave:4> # Contoh statemen kondisional if
```

```
octave:4> function n = nilai(kode)
if kode == "A"
n = 4;
elseif kode == "B"
n = 3;
elseif kode == "C"
n = 2;
elseif kode == "D"
n = 1;
else
n = 0;
end
endfunction
```

```
octave:5> n1 = nilai('B')
n1 = 3
```

```
octave:6> n1 = nilai('E')
n1 = 0
```

```
octave:7> # Contoh kombinasi dari statemen if dan statemen for
```

```
octave:7> I = zeros(3,3);
```

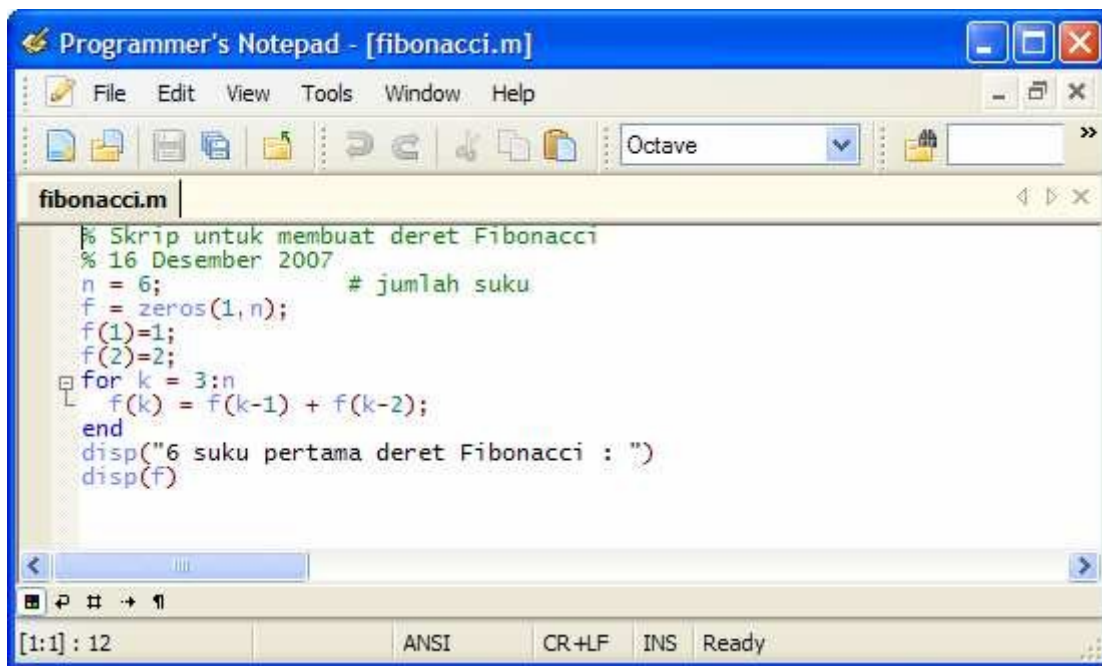
```
octave:8> for i = 1:3
    for j = 1:3
        if (i == j)
            I(i,i) = i;
        end
    end
end
```

```
octave:9> I
I =
1 0 0
0 2 0
0 0 3
```

12. Skrip

Skrip adalah sebuah file teks yang di dalamnya terdapat perintah-perintah Octave. Apabila suatu skrip dijalankan maka perintah-perintah yang terdapat di dalamnya akan dieksekusi seperti seolah-olah kita mengetikkannya pada jendela perintah. Skrip dapat dibuat dengan menggunakan program teks seperti Notepad, Notepad2, Scite dan Programmer's Notepad.

Berikut ini adalah contoh sebuah skrip (`fibonacci.m`) yang menggambarkan perhitungan 6 suku pertama dari deret Fibonacci.



```
Programmer's Notepad - [fibonacci.m]
File Edit View Tools Window Help
Octave
fibonacci.m
% Skrip untuk membuat deret Fibonacci
% 16 Desember 2007
n = 6;          # jumlah suku
f = zeros(1,n);
f(1)=1;
f(2)=2;
for k = 3:n
    f(k) = f(k-1) + f(k-2);
end
disp("6 suku pertama deret Fibonacci : ")
disp(f)
[1:1] : 12  ANSI  CR+LF  INS  Ready
```

Gambar 9

Untuk menjalankan suatu skrip kita dapat menggunakan perintah `run(nama_file)`, dimana `nama_file` adalah nama dari file skrip. Apabila file skrip berada pada direktory Octave (*Octave path*) maka file skrip dapat dijalankan dengan menuliskan nama filenya.

Berikut ini adalah output yang muncul pada jendela perintah jika skrip `fibonacci.m` dijalankan.

```
octave:11> fibonacci
6 suku pertama deret Fibonacci :
1 2 3 5 8 13
```

Untuk melihat daftar direktory Octave, gunakan perintah `path`. Selanjutnya untuk menambahkan suatu direktory ke dalam direktory Octave gunakan perintah `addpath`. Penjelasan yang terperinci mengenai kedua perintah tersebut dan perintah-perintah lainnya yang berkaitan dapat dilihat pada sistem bantuan atau dokumentasi Octave.

13. Fungsi

Fungsi merupakan kumpulan dari statemen-statemen Octave yang dapat melakukan suatu komputasi atau perhitungan tertentu. Fungsi bersifat lebih fleksibel dibanding dengan skrip, karena di dalam fungsi terdapat argumen-argumen input dan output. Fungsi dapat dibuat dalam bentuk suatu file atau secara inline pada jendela perintah. Untuk suatu fungsi yang akan kita gunakan berulang kali maka biasanya fungsi tersebut dibuat dalam bentuk file fungsi, namun untuk suatu fungsi yang hanya digunakan untuk sementara maka kita dapat membuatnya secara inline pada jendela perintah.

Bentuk umum dari suatu fungsi adalah sebagai berikut:

```
function [out1,out2,...] = fname(in1,in2,...)
    statemen-statemen
endfunction
```

dimana `fname` adalah nama fungsi yang kita buat, dan `in1,in2,...` adalah argumen-argumen input serta `out1,out2,...` adalah argumen-argumen output. Nama suatu file fungsi harus sama dengan nama fungsi yang terdapat di dalamnya. File fungsi harus disimpan pada direktory yang termasuk pada *Octave path*.

Contoh pembuatan suatu fungsi secara online adalah sebagai berikut:

```
octave:79> function y = f(x)
    y = x - exp(-x);
endfunction
```

Gambar 10 adalah suatu file fungsi untuk mencari akar dari suatu persamaan nonlinier dengan menggunakan metode Secant.

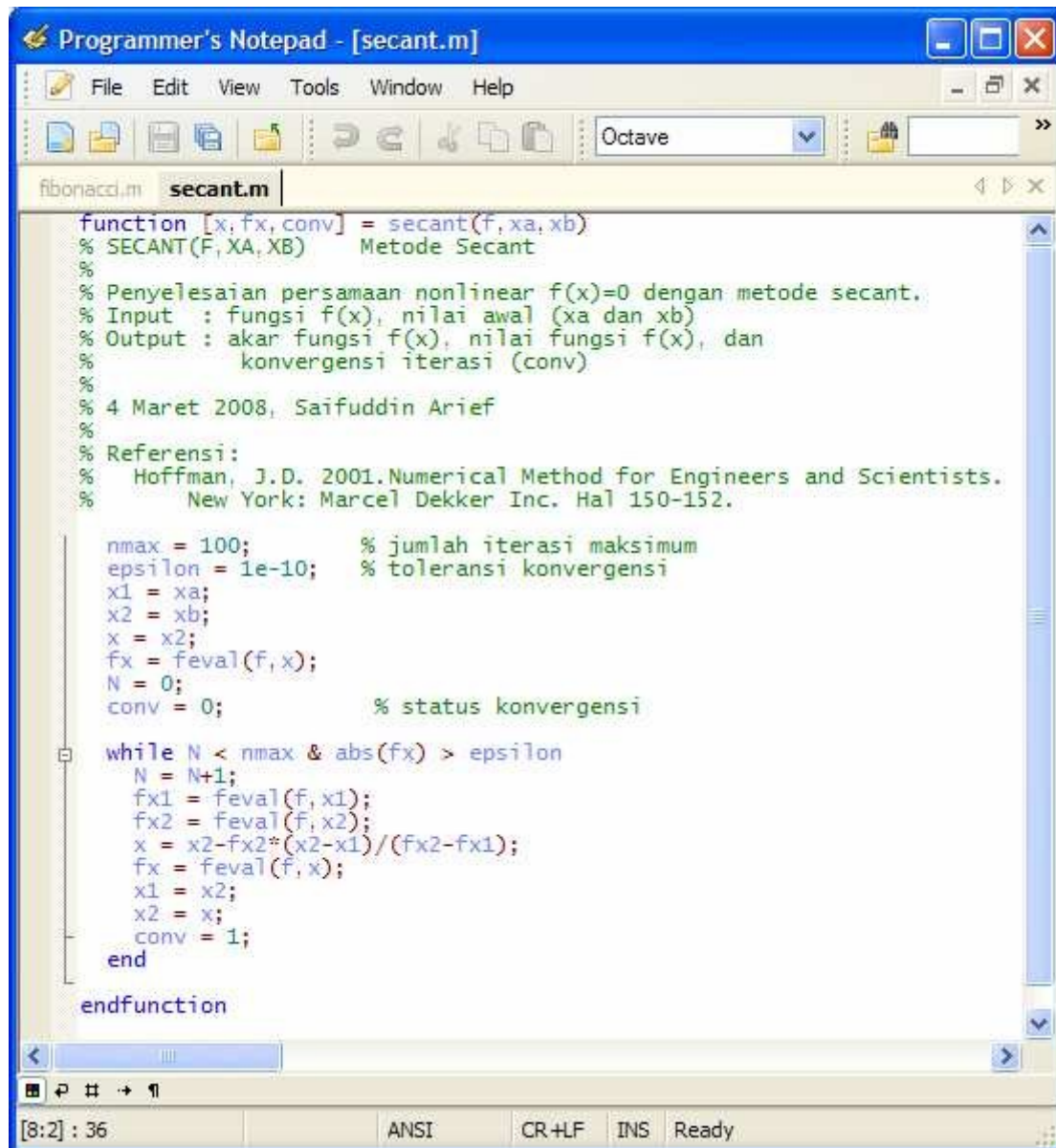
Fungsi-fungsi yang telah kita buat dapat kita gunakan sebagaimana fungsi-fungsi yang telah terpasang pada Octave.

Berikut ini adalah contoh penggunaan fungsi secant untuk mencari akar dari fungsi yang telah kita definisikan di atas.

```
octave:80> [x0, fx0, conv] = secant("f",1,2)
x0 = 0.56714
fx0 = 5.0341e-011
conv = 1
```

Argumen input untuk fungsi "f" Statemen di atas juga dapat dinyatakan dengan menggunakan cara lain yaitu `@f` seperti pada contoh di bawah ini.

```
octave:81> [x0, fx0, conv] = secant(@f,1,2)
x0 = 0.56714
fx0 = 5.0341e-011
conv = 1
```



```
function [x,fx,conv] = secant(f,xa,xb)
% SECANT(F,XA,XB)   Metode Secant
%
% Penyelesaian persamaan nonlinear f(x)=0 dengan metode secant.
% Input  : fungsi f(x), nilai awal (xa dan xb)
% Output : akar fungsi f(x), nilai fungsi f(x), dan
%         konvergensi iterasi (conv)
%
% 4 Maret 2008, Saifuddin Arief
%
% Referensi:
% Hoffman, J.D. 2001.Numerical Method for Engineers and Scientists.
% New York: Marcel Dekker Inc. Hal 150-152.

nmax = 100;      % jumlah iterasi maksimum
epsilon = 1e-10; % toleransi konvergensi
x1 = xa;
x2 = xb;
x = x2;
fx = feval(f,x);
N = 0;
conv = 0;       % status konvergensi

while N < nmax & abs(fx) > epsilon
    N = N+1;
    fx1 = feval(f,x1);
    fx2 = feval(f,x2);
    x = x2-fx2*(x2-x1)/(fx2-fx1);
    fx = feval(f,x);
    x1 = x2;
    x2 = x;
    conv = 1;
end

endfunction
```

Gambar 10

Salah satu kelebihan Octave yaitu dalamnya telah terpasang sejumlah besar fungsi-fungsi matematika dan algoritma-algoritma metode numerik. Hal tersebut dapat terwujud disebabkan oleh adanya fasilitas pemrograman dan file fungsi yang disediakan oleh Octave.

Dengan menggunakan fasilitas pemrograman dan file fungsi kita juga dapat menambahkan suatu fungsi matematika atau algoritma metode numerik yang belum terdapat di dalam Octave, sesuai dengan yang kita perlukan.

14. Daftar Pustaka

1. Chapra, S.C., Canale, R.P., Numerical Methods for Engineers with Programming and Software Applications. WCB/McGraw-Hill, Singapore, 1998.
2. Eaton, J. W., GNU Octave Manual, Network Theory Limited, 2002.
3. Octave, <http://www.octave.org>.