

Pengenalan FreeMat

Saifuddin Arief

Saifuddin.Arief@rocketmail.com

Lisensi Dokumen:

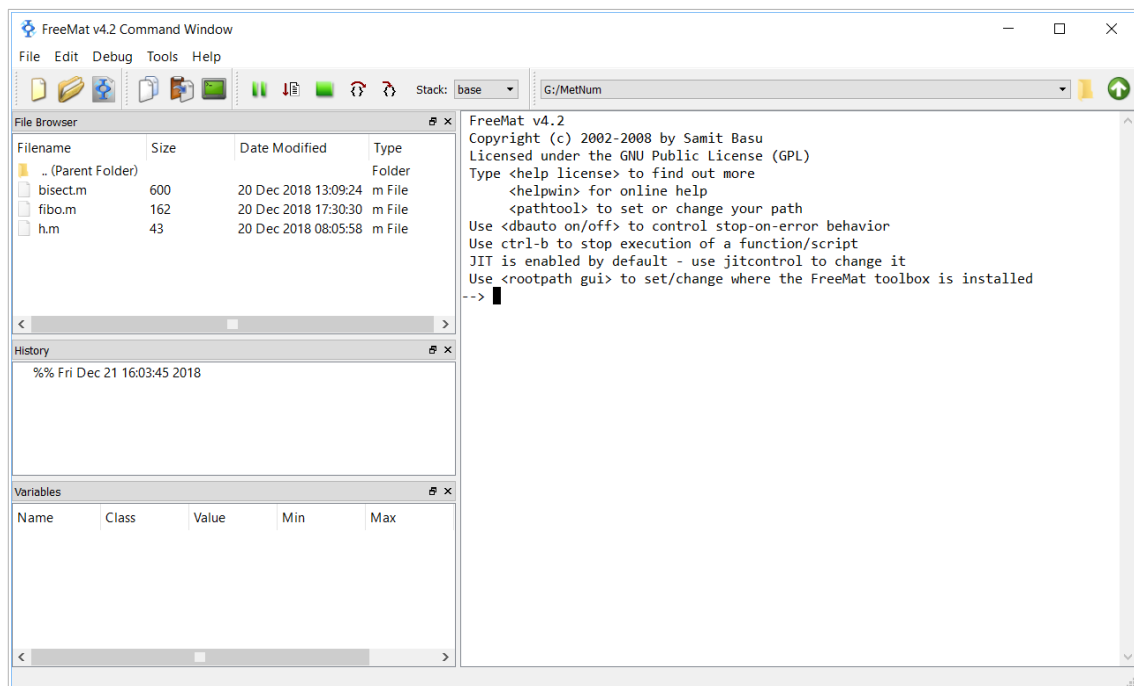
Copyright © 2003-2018 IlmuKomputer.Com

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.

FreeMat adalah *freeware* yang dikembangkan untuk komputasi numerik. FreeMat dirancang sebagai alternatif dari Matlab. Alamat situs FreeMat adalah <http://freemat.sourceforge.net/>.

Menjalankan program FreeMat

FreeMat dapat dijalankan melalui menu utama atau melalui terminal dengan menggunakan perintah `freemat`. Setelah FreeMat dijalankan maka pada layar komputer akan muncul jendela FreeMat seperti yang terlihat pada Gambar 1.



Gambar 1. FreeMat

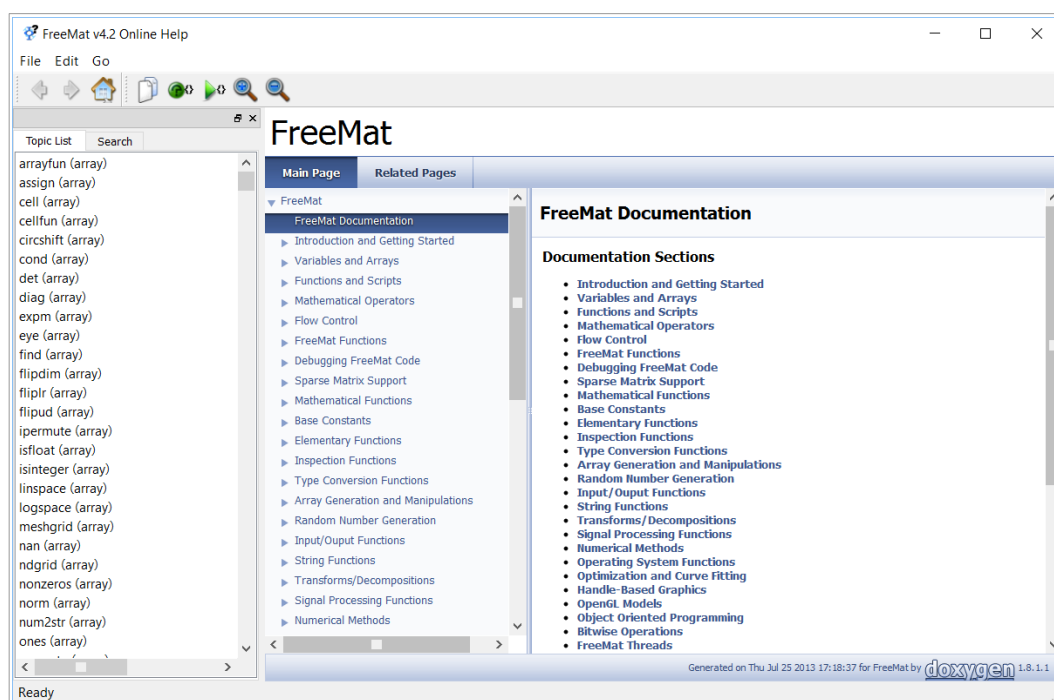
Simbol `-->` yang terdapat pada jendela FreeMat merupakan tanda bahwa FreeMat siap untuk menerima suatu perintah. Sebagai contoh, untuk melakukan perhitungan $1.23 + 4.56$ maka

ekspresi matematika tersebut diketikkan setelah simbol `-->` kemudian ditekan tombol `enter` untuk menjalankan perintahnya. Hasil perhitungannya akan ditampilkan pada baris berikutnya.

```
--> 1.23 + 4.56  
ans =  
    5.7900
```

Tanda `-->` yang muncul kembali menunjukkan bahwa FreeMat siap untuk mengerjakan suatu perintah yang lain.

Penjelasan dan dokumentasi mengenai penggunaan program FreeMat dapat dilihat pada jendela Dokumentasi. Jendela tersebut dapat ditampilkan melalui menu `Help - Online Manual` atau dengan menjalankan perintah `helpwin`. Ilustrasi dari jendela dokumentasi diberikan pada gambar di bawah ini.



Gambar 2. Jendela Bantuan FreeMat

Untuk keluar dari program FreeMat dapat dilakukan dengan menggunakan perintah `exit` atau `quit`, melalui menu `File - Exit` atau dengan menekan tombol `×` yang terletak pada bagian kanan atas dari jendela FreeMat.

Variabel dan Ekspresi

Variabel adalah sebuah nama yang digunakan untuk menyimpan nilai suatu obyek. Penyimpanan nilai ke dalam suatu variabel dilakukan dengan dengan statemen sebagai berikut:

```
x = ekspresi
```

dimana `x` adalah nama variabel. Dalam penggunaan variabel, huruf kecil dan huruf besar adalah berbeda. Nilai dari suatu ekspresi akan ditampilkan pada baris berikutnya, kecuali jika ditambahkan tanda titik koma (`;`) pada akhir ekspresinya.

Di dalam FreeMat terdapat beberapa variabel khusus yang menyatakan suatu konstanta matematika, seperti `pi` untuk $\pi = 3.1415927\dots$, dan `e` untuk $e = 2.7182818\dots$

Untuk memperjelas perintah yang dibuat, dapat ditambahkan suatu baris komentar. Baris komentar dapat ditulis sebagai suatu baris tersendiri atau ditulis di belakang suatu statemen. Kumpulan karakter yang terletak setelah tanda % akan dianggap sebagai baris komentar.

```
--> luas = 12.5*8
luas = 100

--> r = 10;          % radius lingkaran
--> A = pi*r^2      % luas lingkaran
A = 314.16
```

Untuk suatu ekspresi cukup panjang dan tidak cukup untuk ditulis pada satu baris maka digunakan tanda titik tiga (...) sebagai tanda sambung dengan baris berikutnya.

```
--> e6 = 1 + 1/1 + 1/(1*2) + 1/(1*2*3) + 1/(1*2*3*4) + 1/(1*2*3*4*5) + ...
1/(1*2*3*4*5*6)
e6 = 2.7181
```

Ruang Kerja

Variabel-variabel yang telah dibuat akan disimpan dalam ruang kerja. Untuk melihat nama-nama variabel yang telah dibuat, gunakan perintah `who`. Variabel yang tersimpan dalam ruang kerja dapat dihapus dengan perintah `clear`.

```
--> who
Variable Name      Type      Flags      Size
      A      double
      ans     double
      e6      double
      luas    double
      r      double

--> clear luas r          % menghapus variabel luas dan r

--> clear                % menghapus semua variabel
```

Operator-operator dan Fungsi-fungsi Matematika

Operator-operator untuk perhitungan aritmatika yaitu +, -, *, / dan ^. Simbol tersebut masing-masing melambangkan operasi penjumlahan, pengurangan, perkalian, pembagian dan pemangkatan. Di dalam FreeMat juga telah terpasang fungsi-fungsi matematika, seperti `sqrt`, `abs`, `exp`, `sin`, `cos`, `tan` dan lain sebagainya.

```
--> 1010*(sqrt(9.81^2 + 7^2)*((0.07 + 0.0214)*cos(35.5/180*pi)))
ans = 905.7136

--> k = 0.08567/(pi*(5.8^2 - 4.5^2))*log(25/3.5)
k = 0.0040
```

Bilangan Kompleks

Bilangan kompleks $z = x + iy$ dinyatakan dengan notasi $z = x + y*i$ atau $z = x + y*j$, dimana i dan j adalah variabel khusus yang merepresentasikan konstanta $\sqrt{-1}$. Operasi aritmatika terhadap bilangan kompleks dapat dilakukan dengan menggunakan notasi yang sama dengan notasi pada bilangan real.

```
--> z1 = 6 - 8*i;

--> z2 = 3 + i;
```

```
--> p = z1 + z2
p = 9 - 7i

--> q = z1 - z2
q = 3 - 9i

--> z1*z2
ans = 26 - 18i

--> z2/z1
ans = 0.10000 + 0.30000i
```

Matrik dan Vektor

Pada dasarnya semua data numerik di dalam FreeMat 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.

Data matrik dan vektor dibuat dengan menggunakan operator kurung siku ([]). Untuk memisahkan elemen yang satu dengan elemen yang lainnya yang terletak pada satu baris dapat digunakan tanda koma (,) atau tanda spasi. Kemudian antara baris yang satu dengan yang lainnya dipisahkan dengan tanda titik koma (;) atau tombol Enter.

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

--> v = [1 2 3 4 5];           % vektor baris

--> w = [3;4;5]               % vektor kolom
w =
  3
  4
  5
```

Suatu vektor baris dimana nilai elemen-elemennya berubah secara konstan dari suatu nilai awal sampai nilai akhir tertentu dapat dibuat dengan sintak $i:j:k$. Apabila nilai j sama satu maka notasi tersebut dapat ditulis dengan notasi yang lebih singkat yaitu $i:k$.

```
--> i = 1:10
i =
  1  2  3  4  5  6  7  8  9 10

--> n = 0:0.25:1
n =
  0.00000  0.25000  0.50000  0.75000  1.00000
```

FreeMat juga menyediakan sejumlah fungsi yang dapat digunakan untuk membuat matrik-matrik khusus, seperti yang diilustrasikan pada contoh-contoh di bawah ini.

```
--> A = zeros(3,4)           % matrik nol
A =
  0  0  0  0
  0  0  0  0
  0  0  0  0
```

```
--> B = ones(2,5)           % matrik satuan
B =
    1    1    1    1    1
    1    1    1    1    1

--> Y = eye(3)              % matrik identitas
Y =
    1    0    0
    0    1    0
    0    0    1

--> D = diag(1:4)          % matrik diagonal
D =
    1    0    0    0
    0    2    0    0
    0    0    3    0
    0    0    0    4
```

Operasi Berbasis Vektor

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

```
--> x = 0:pi/4:pi;

--> cos(x)
ans =
    1.0000    0.7071    0.0000   -0.7071   -1.0000

--> u = [1 2 3];

--> y = exp(u)
y =
    2.7183    7.3891   20.0855

--> z = log(y)
z =
    1    2    3
```

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 dimensinya tidak kompatibel maka operasinya tidak dapat dieksekusi dan akan muncul suatu pesan kesalahan.

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

--> Y = [3 2 1; 4 5 6; 9 8 7];

--> A = X + Y
A =
    12    10     6
     5     8     6
    11    12    13

--> B = X - Y
B =
     6     6     4
    -3    -2    -6
    -7    -4    -1

--> p = [1 2; 3 8];
```

```
--> Z = A + p
Error: size mismatch in arguments to binary operator

--> j = [6;4];

--> k = ones(3);

--> g = p*j
g =
    14
    50

--> h = p*k
Error: Requested matrix multiplication requires arguments to be conformant
```

Selain operasi perkalian yang mengikuti aturan dalam Aljabar Linier, dapat juga dilakukan 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 `./`.

```
--> x = [1 2; 3 4];

--> y = [5 6; 7 8];

--> u = x.*y
u =
     5    12
    21    32

--> z = y./x
z =
     5.0000     3.0000
     2.3333     2.0000
```

Perulangan dan Kondisional

FreeMat menyediakan sejumlah kontrol pemrograman yang dapat digunakan untuk mengatur jalannya eksekusi suatu program dengan menggunakan statemen perulangan dan kondisional. Ilustrasi penggunaan statemen perulangan `for` dan kondisional `if` diberikan pada contoh-contoh di bawah ini.

```
--> H = zeros(4,4);

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

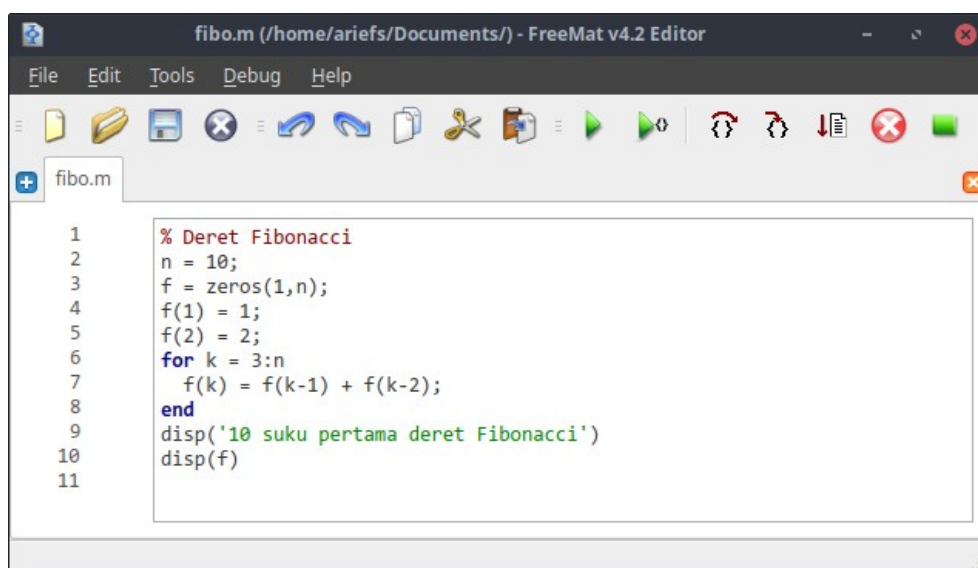
--> H
H =
    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429

--> function n=nilai(kode)
    # Fungsi untuk mengkonversi nilai dari abjad menjadi angka
    if kode=='A'
        n=4;
    elseif kode=='B'
        n=3;
```

```
elseif kode=='C'  
    n=2;  
elseif kode=='D'  
    n=1;  
else  
    n=0;  
end  
  
--> n1 = nilai('A')  
n1 = 4
```

Skrip

Skrip adalah sebuah file teks yang di dalamnya terdapat perintah-perintah FreeMat. Apabila suatu skrip dijalankan maka perintah-perintah yang terdapat di dalamnya akan dieksekusi seperti seolah-olah kita mengetikkannya pada jendela perintah. Editor adalah fitur yang disediakan oleh FreeMat untuk pembuatan sebuah skrip atau file fungsi. Berikut ini adalah contoh sebuah skrip (`fibonacci.m`) yang menggambarkan perhitungan 10 suku pertama dari deret Fibonacci.



Gambar 5.

Statemen-statemen yang terdapat di dalam sebuah skrip yang sedang dibuka pada jendela Editor dapat dijalankan melalui menu **Debug**. Jika skrip `fibonacci.m` dijalankan maka tampilan berikut ini akan muncul pada Jendela FreeMat.

```
--> fibo  
  
10 suku pertama deret Fibonacci :  
    1    2    3    5    8   13   21   34   55   89
```

Sebuah file skrip yang tersimpan pada sebuah direktori kerja atau direktori yang terbaca oleh FreeMat dapat dieksekusi atau dijalankan dengan statemen `fskrip`. Apabila filenya tidak tersimpan pada direktori tersebut maka dapat dijalankan dengan perintah `source fskrip` dimana `fskrip` nama file skrip dan nama direktorinya.

Fungsi

Fungsi merupakan kumpulan dari statemen-statemen FreeMat yang dapat melakukan suatu komputasi atau perhitungan tertentu. Bentuk umum dari suatu fungsi adalah sebagai berikut:

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

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 direktori yang termasuk pada FreeMat *path*.

Suatu file-fungsi yang telah selesai dibuat dengan editor dapat disimpan melalui menu File - Save. Ilustrasi dari sebuah file fungsi diberikan pada gambar 6. Fungsi tersebut [`bisect.m`] adalah fungsi untuk mencari akar dari suatu persamaan nonlinier dengan metode *bisection*.

```
bisect.m (E:/MetNum/) - FreeMat v4.2 Editor
File Edit Tools Debug Help
bisect.m
1 function [x, fx] = bisect(f, x1, x2, tol)
2 % Penentuan akar persamaan nonlinier dengan metode Bisection
3
4 if f(x1) == 0.0; x = x1; return; end
5 if f(x2) == 0.0; x = x2; return; end
6 if f(x1)*f(x2) > 0
7     error('Akar tidak terdapat di dalam selang [x1,x2]')
8 end
9
10 n = ceil(log(abs(x2 - x1)/tol)/log(2.0)); % Iterasi maksimum
11
12 for i = 1:n
13     x3 = 0.5*(x1 + x2);
14     f3 = f(x3);
15     if f3 == 0.0
16         x = x3; return
17     end
18     if f(x2)*f(x3) < 0.0
19         x1 = x3; f1 = f3;
20     else
21         x2 = x3; f2 = f3;
22     end
23 end
24 x = (x1 + x2)/2;
25 fx = f(x);
26
27 endfunction
```

Gambar 6.

Untuk sebuah fungsi matematika yang sederhana dapat dibuat secara inline sebagai fungsi anonim dengan notasi `y = @(arg1,arg2,...) ekspresi`, seperti pada contoh di bawah ini.

```
--> fx = @(x) (x - exp(-x));
```

Berikut ini adalah contoh penggunaan fungsi `bisect` dan `fx` yang telah dibuat.

```
--> fx(1)
ans =
    0.6321
```



```
--> [x0,fx0] = bisect(fx,0,1,1e-6)
x0 =
    0.5671
fx0 =
-5.1246e-07
```

Komputasi Aljabar Linier

Di dalam FreeMat, penyelesaian berbagai macam persoalan dalam komputasi aljabar linier dapat dilakukan dengan mudah.

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

```
--> 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
```

```
--> b = [13; 28; 20; 6]
b =
```

```
    13
    28
    20
     6
```

```
--> x = A \ b
x =
```

```
    3.0000
   -1.0000
    4.0000
    2.0000
```

Nilai determinan dari matrik A dapat dihitung dengan menggunakan fungsi $\text{det}(A)$ dan $\text{inv}(A)$.

```
--> det(A)
ans = -180
```

```
--> inv(A)
ans =
```

```
-0.000000  0.083333  0.083333 -0.16667
 0.066667 -0.34444  0.32222  0.22222
-0.200000  0.11667  0.11667  0.16667
 0.26667  0.12222 -0.21111 -0.11111
```

Daftar Pustaka

Basu, S. 2009. FreeMat v4.0 Documentation.
Schafer, G., Cyders, T. 2011. The Freemat 4.0 Primer.