

Pengantar Komputasi Numerik dengan Mathnium

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.

Mathnium merupakan perangkat lunak yang dikembangkan untuk komputasi numerik dan visualisasi data secara interaktif. Mathnium dikembangkan dalam bahasa Java sehingga dapat dijalankan sistem operasi Windows dan Linux. Fungsi-fungsi yang terdapat di dalamnya relatif sedikit namun sudah mencukupi karena fungsi-fungsi untuk metode numerik yang sering digunakan sudah diimplementasikan. Alamat situs Mathnium adalah <http://www.mathnium.com/>.

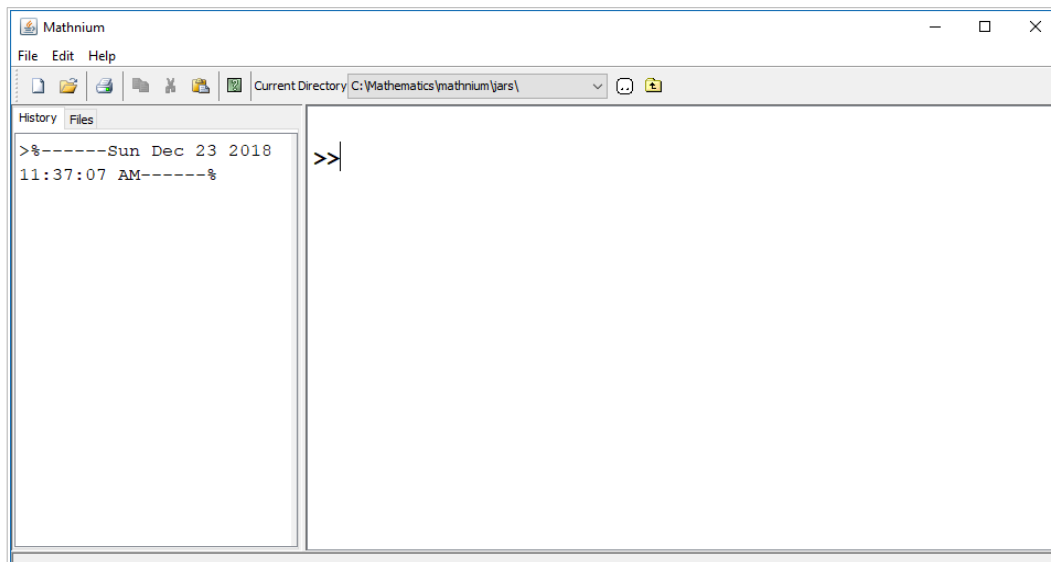
Menjalankan program Mathnium

Mathnium dapat dijalankan melalui terminal dengan menjalankan file `math.bat` pada Windows atau `math.sh` pada Linux. Setelah Mathnium dijalankan maka pada layar komputer akan muncul jendela Mathnium seperti yang terlihat pada Gambar 1.

Simbol `>>` yang terdapat pada jendela Mathnium merupakan tanda bahwa Mathnium siap untuk menerima suatu perintah. Sebagai contoh, perhitungan $1.23 + 4.56$ dilakukan dengan mengetikkan ekspresi matematika tersebut setelah simbol `>>` kemudian menekan tombol Enter untuk menjalankan perintahnya. Hasil perhitungannya akan ditampilkan pada baris berikutnya.

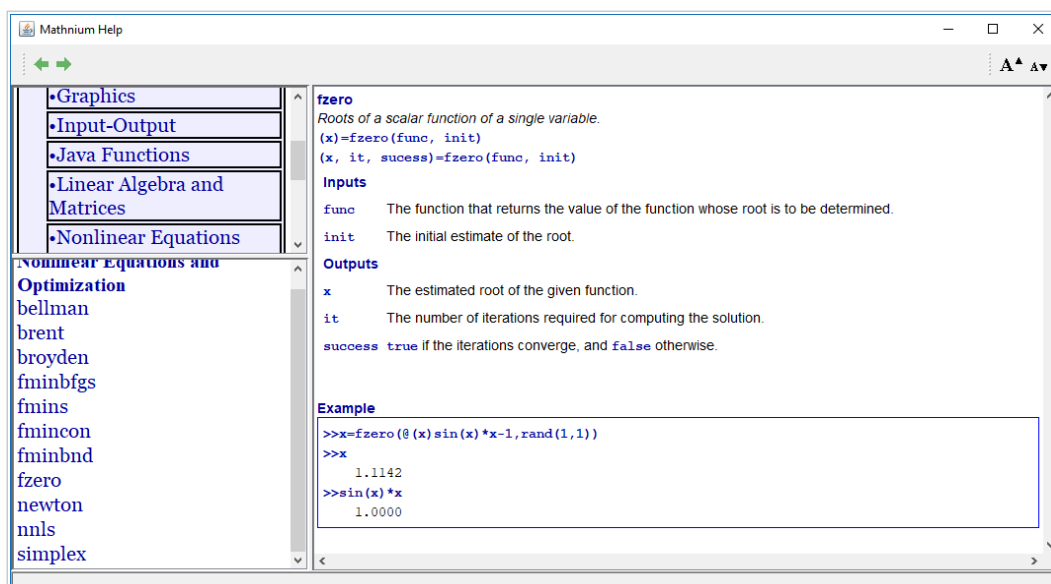
```
>> 1.23 + 4.56  
ans = 5.7900
```

Tanda `>>` yang muncul kembali menunjukkan bahwa Mathnium siap untuk menerima suatu perintah yang lain.



Gambar 1. Mathnium

Penjelasan dan dokumentasi mengenai penggunaan program Mathnium dapat dilihat pada jendela Dokumentasi. Jendela tersebut dapat ditampilkan melalui menu **Help - Help**.



Gambar 2. Jendela Bantuan Mathnium

Perintah **help** dapat digunakan untuk mendapatkan penjelasan mengenai suatu perintah, fungsi atau operator tertentu. Sebagai contoh, penjelasan tentang fungsi **abs** dapat diperoleh dengan menjalankan perintah **help abs**.

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

Variabel dan Ekspresi

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

```
x = ekspresi
```

dimana x adalah nama variabel. Dalam penggunaan variabel, huruf kecil dan huruf besar adalah berbeda. Nilai suatu variabel dapat dilihat dengan menjalankan nama variabelnya.

Di dalam Mathnium 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 // atau terletak diantara /* dan */ akan dianggap sebagai baris komentar.

```
>>area = lebar*tinggi
>>area
           100

>>r = 10           // radius lingkaran

>>A = pi*r^2      // luas lingkaran
>>A
           314.1593
```

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
```

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 Mathnium 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)))
           905.7136

>>k = 0.08567/(pi*(5.8^2 - 4.5^2))*log(25/3.5)
>>k
           0.004
```

Bilangan Kompleks

Bilangan kompleks $z = x + iy$ dinyatakan dengan notasi $z = x + yi$, dimana i 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
          26 - 18i
>>z2/z1
          0.1 + 0.3i
```

Matrik dan Vektor

Salah satu kelebihan Mathnium yaitu kemampuannya dalam menangani berbagai macam operasi manipulasi terhadap data yang berupa 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 dapat 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]
>>v = [1 2 3 4 5] // vektor baris
>>w = [3;4;1;2] // vektor kolom
```

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 = 1:6
>>i
          1          2          3          4          5          6
>>n = 0:0.25:1
>>n
          0          0.25          0.5          0.75          1
>>m = 1:2:10
>>m
          1          3          5          7          9
```

Mathnium 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(2)            // matrik identitas
>>Y
      1      0
      0      1

>>D = diag(1:3)        // matrik diagonal
>>D
      1      0      0
      0      2      0
      0      0      3
```

Operasi Berbasis Vektor

Di dalam Mathnium, secara umum operasi-operasi matematika terhadap obyek matrik dan vektor dapat dilakukan dengan mudah tanpa harus menggunakan suatu perulangan.

```
>>x = 0:pi/4:pi
>>cos(x)
      1      0.7071  6.1232e-017  -0.7071  -1

>>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, dapat dilakukan dengan mudah tanpa harus menggunakan suatu ekspresi perulangan. 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]
>>X + Y
      12      10      6
      5      8      6
      11      12      13

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

>>p = [1 2; 3 8]
>>X + p
Error Invalid operand sizes for matrix addition/subtraction

>>j = [6;4]
```

```
>>k = ones(3)
```

```
>>p*j
    14
    50
```

```
>>p*k
Error Invalid operand sizes for matrix multiplication/division
```

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]
```

```
>>x.*y
    5    12
   21    32
```

```
>>y./x
    5    3
   2.3333 2
```

Perulangan dan Kondisional

Mathnium menyediakan sejumlah kontrol pemrograman yang dapat digunakan untuk mengatur jalannya eksekusi suatu program dengan menggunakan statemen perulangan `for` dan kondisional `if-else`. Ilustrasi penggunaan statemen perulangan dan kondisional 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
```

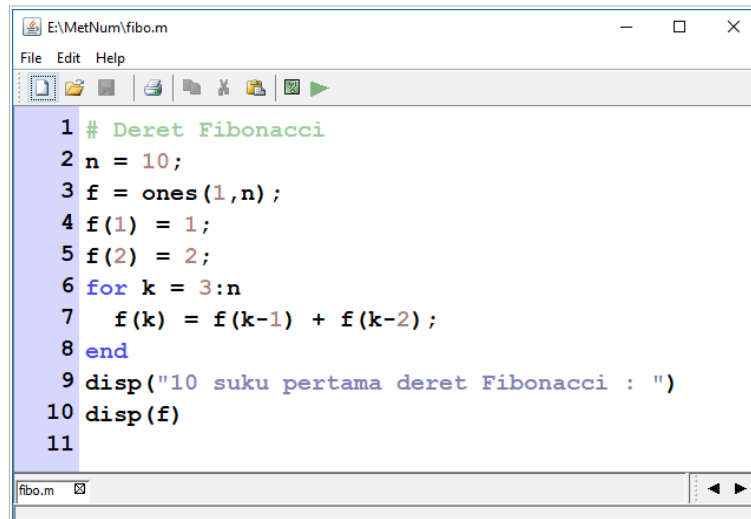
```
>>H
    1    0.5    0.3333    0.25
    0.5    0.3333    0.25    0.2
    0.3333    0.25    0.2    0.1667
    0.25    0.2    0.1667    0.1429
```

```
>>function n=nilai(kode)
// Fungsi untuk mengkoversi 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
end
```

```
>>nilai('A')  
4
```

Skrip

Skrip adalah sebuah file teks yang di dalamnya terdapat perintah-perintah Mathnium. Apabila suatu skrip dijalankan maka perintah-perintah yang terdapat di dalamnya akan dieksekusi seperti seolah-olah kita mengetikkannya pada jendela perintah. Untuk membuat sebuah file skrip dapat dilakukan melalui menu File - New. Gambar 5 adalah contoh sebuah skrip (fibonacci.m) yang menggambarkan perhitungan 10 suku pertama dari deret Fibonacci.



```
E:\MetNum\fibonacci.m  
File Edit Help  
1 # Deret Fibonacci  
2 n = 10;  
3 f = ones(1,n);  
4 f(1) = 1;  
5 f(2) = 2;  
6 for k = 3:n  
7 f(k) = f(k-1) + f(k-2);  
8 end  
9 disp('10 suku pertama deret Fibonacci : ')  
10 disp(f)  
11
```

Gambar 4.

Statemen-statemen yang terdapat di dalam sebuah skrip yang sedang dibuka pada jendela Editor dapat dijalankan melalui ikon ▶. Jika skrip fibonacci.m dijalankan maka tampilan berikut ini akan muncul pada jendela Mathnium.

```
>> 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 Mathnium dapat dieksekusi atau dijalankan dengan statemen `fskrip` dimana `fskrip` adalah nama filenya. Apabila filenya tidak tersimpan pada direktori-direktori tersebut maka argumen `fskrip` harus berupa nama file skrip dan nama direktorinya.

Fungsi

Fungsi merupakan kumpulan dari statemen-statemen Mathnium yang dapat melakukan suatu komputasi atau perhitungan tertentu. Fungsi dapat dibuat dalam bentuk suatu file atau secara inline pada jendela perintah. Bentuk umum dari suatu fungsi adalah sebagai berikut:

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

dimana `fname` adalah nama fungsi yang dibuat, 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. Ilustrasi dari sebuah file fungsi diberikan pada gambar 6. Fungsi tersebut [`bisect.m`] adalah fungsi untuk mencari akar dari suatu persamaan nonlinear dengan metode *bisection*.

```

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

Gambar 5.

Contoh pembuatan suatu fungsi secara inline adalah sebagai berikut:

```

>>function y = fx(x)
> y = x - exp(-x)
> endfunction
    
```

Fungsi yang telah dibuat dapat digunakan sebagaimana fungsi-fungsi yang telah terpasang pada Mathnium. Berikut ini adalah contoh penggunaan fungsi `bisect` dan `f` yang telah dibuat.

```

>>fx(0)
        -1

>>(x0,fx0) = bisect(@fx,0,1,1e-6)
>>x0,fx0
        0.5671        -5.1246e-007
    
```

Komputasi Numerik

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

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

```

>>A = [1 2 1 4; 2 0 4 3; 4 2 2 1; -3 1 3 2]
>>b = [13; 28; 20; 6]
    
```



```
>>x = A\b
>>x
      3
     -1
      4
      2
```

Nilai determinan dan inverse matrik A dapat dihitung dengan fungsi `det(A)` dan `inv(A)`.

```
>>det(A)
-180

>>inv(A)
-6.9389e-018    0.0833    0.0833    -0.1667
      0.0667    -0.3444    0.3222    0.2222
      -0.2     0.1167    0.1167    0.1667
      0.2667    0.1222   -0.2111   -0.1111
```

Penyelesaian suatu persamaan nonlinier $f(x) = 0$ dapat dilakukan dengan fungsi `fzero`, seperti yang ditunjukkan pada contoh di bawah ini.

```
>>function y = fx(x)
> y = x - exp(-x)
> end

>>(x0,it,kond) = fzero(@fx,0.5)
>>x0,it,kond
      0.5671      4      1
```

Diperoleh bahwa akar dari $f(x) = x - \exp(-x) = 0$, adalah $x = 0.5671$.

Terdapat banyak persamaan integral tertentu yang sulit atau bahkan tidak dapat diselesaikan secara analitis, salah satunya adalah seperti yang terdapat pada contoh berikut ini.

$$Q = \int_0^5 \frac{\sin(3x)}{\sqrt{x^2 + x + 1}} dx$$

Nilai suatu integral tertentu dapat dihitung dengan menggunakan fungsi `quad`. Misalkan $h(x)$ adalah fungsi yang diintegrasikan pada persamaan di atas:

```
>>function y = h(x)
> y = sin(3*x)/sqrt(x^2 + x + 1)
> end
```

Selanjutnya, nilai integral dapat dihitung dengan perintah sebagai berikut.

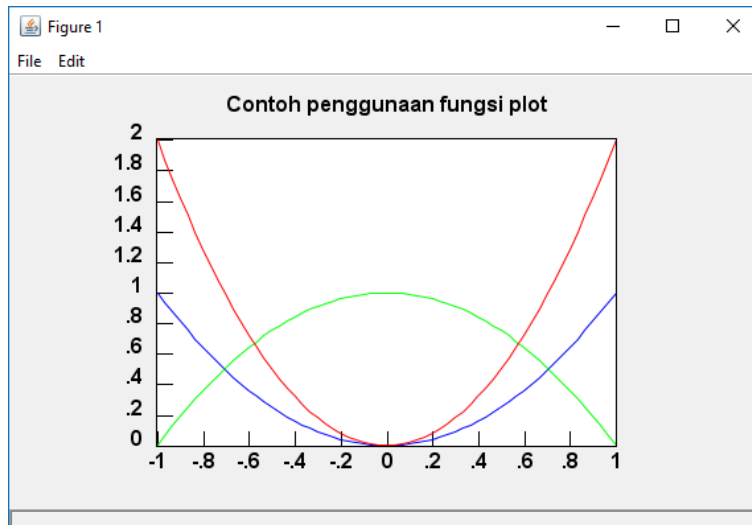
```
>>Q = quad(@h, 0, 5)
>>Q
      0.3649
```

Jawaban yang diperoleh yaitu $Q = 0.3649$.

Visualisasi Data

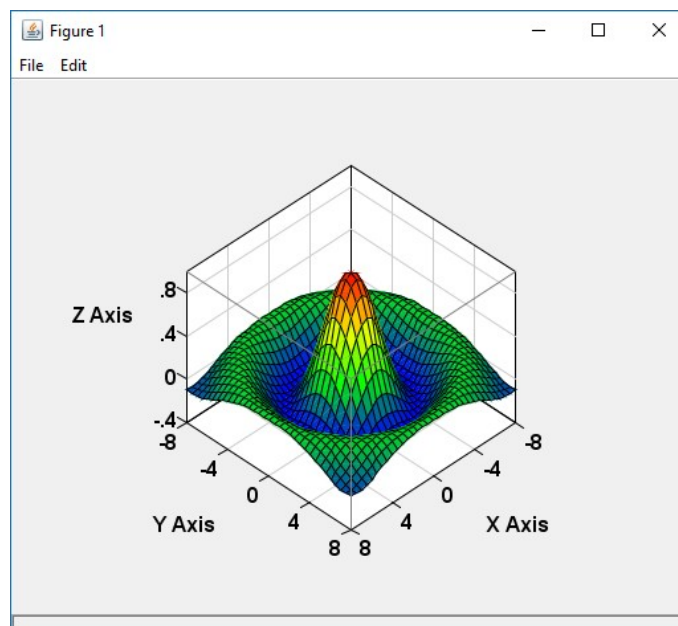
Mathnium dapat digunakan untuk melakukan visualisasi data, baik secara dua dimensi maupun tiga dimensi. Untuk membuat grafik dua dimensi dapat digunakan perintah `plot`, kemudian untuk grafik tiga dimensi dapat digunakan perintah `surf`.

```
>> x = linspace(-1,1,61)';  
>> y1 = x.^2; y2 = 1 - y1; y3 = 2*y1;  
>> plot(x,[y1 y2 y3]) // (lihat Gambar 3)  
>> title('Contoh penggunaan fungsi plot')
```



Gambar 3. Contoh penggunaan fungsi plot

```
>>(x,y) = meshgrid(-8:0.5:8,-8:0.5:8)  
>>r = sqrt(x.^2 + y.^2) + eps  
>>z = sin(r)./r  
>>surf(x,y,z) // (lihat Gambar 4)
```



Gambar 4. Contoh penggunaan fungsi surf