

---

Nama : Rofi'ah 'Aisy Zukhruf  
NIM : 23030630019  
Kelas : Matematika B

---

## Menggambar Grafik 2D dengan EMT

---

Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi `plot2d()` untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

### Plot Dasar

---

Ada fungsi yang sangat mendasar dari plot. Ada koordinat layar, yang selalu berkisar dari 0 hingga 1024 di setiap sumbu, tidak peduli apakah layarnya persegi atau tidak. Semut ada koordinat plot, yang dapat diatur dengan `setplot()`. Pemetaan antara koordinat tergantung pada jendela plot saat ini. Misalnya, `shrinkwindow()` default menyisakan ruang untuk label sumbu dan judul plot.

Dalam contoh, kita hanya menggambar beberapa garis acak dalam berbagai warna. Untuk detail tentang fungsi ini, pelajari fungsi inti EMT.

```
>clg; // clear screen
>window(0,0,1024,1024); // use all of the window
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2); // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // end overwrite mode
>insimg; // insert to notebook
```



---

### Catatan Baru :

1. `clf;` : Perintah ini digunakan untuk membersihkan layar atau area plot yang ada, sehingga siap untuk grafik baru.
  2. `window(0,0,1024,1024);` Perintah ini mengatur dimensi jendela plot dari koordinat (0, 0) hingga (1024, 1024), sehingga menggunakan seluruh ruang jendela.
  3. `setplot(0,1,0,1);` Ini mengatur sistem koordinat untuk plot, di mana sumbu x berkisar dari 0 hingga 1 dan sumbu y juga dari 0 hingga 1.
  4. `hold on;` Perintah ini memungkinkan beberapa plot digambar pada grafik yang sama tanpa menghapus plot sebelumnya.
  5. `n=100; X=random(n,2); Y=random(n,2);` Di sini, 'n' diatur menjadi 100, dan dua array, 'X' dan 'Y', dibuat dengan nilai acak. Masing-masing array berisi 100 titik (koordinat 2D).
  6. `colors=rgb(random(n),random(n),random(n));` Perintah ini menghasilkan array warna acak untuk setiap dari 100 titik. Fungsi 'rgb' kemungkinan membuat nilai warna berdasarkan komponen RGB acak.
  7. `loop 1 to n; color(colors[i]); plot(X[i],Y[i]); end;` Loop ini mengiterasi setiap titik (dari 1 hingga 100), mengatur warna untuk setiap titik berdasarkan array 'colors', dan menggambar koordinat yang sesuai dari 'X' dan 'Y'.
  8. `hold off;` Perintah ini mengakhiri mode overwrite, artinya plot berikutnya akan menghapus grafik yang sedang ditampilkan.
  9. `insimg;` Akhirnya, perintah ini kemungkinan menyisipkan gambar atau plot yang dihasilkan ke dalam notebook atau dokumen untuk digunakan atau analisis lebih lanjut.
- Secara keseluruhan, kode ini menghasilkan plot sebar 100 titik acak dengan warna acak dalam sistem koordinat yang ditentukan, menampilkannya, dan menyiapkannya untuk dokumentasi.

---

### contoh penyelesaian masalah menggambar kurva/plot 2D

---

1.

---

## Menggambar Fungsi Kuadrat

$$f(x) = x^2$$

### Penyelesaian :

---

```
>clc; // clear screen
>window(-10,-10,10,10); // set window for x and y
>setplot(-10,10,0,100); // set plot coordinates for the function
```

```
>hold on; // start overwrite mode
>// Definisikan fungsi kuadrat
>n=200; // jumlah titik
>x=random(n)*20-10; // menghasilkan n titik x antara -10 hingga 10
>y=x.^2; // hitung y untuk f(x) = x^2
```

```
>// Gambar kurva
>plot(x,y,"b"); // plot titik (x,y) dengan warna biru
```

```
Function plot needs only 2 arguments (got 3)!
Use: plot ( {x, y})
Error in:
plot(x,y,"b"); // plot titik (x,y) dengan warna biru ...
      ^
```

```
>hold off; // end overwrite mode
```

```
>insimg; // insert to notebook
```

Penjelasan Kode:

lg: Membersihkan layar.

indow: Mengatur batas jendela untuk tampilan.

etplot: Menentukan koordinat plot.

old on: Mengizinkan gambar bertumpuk.

andom: Menghasilkan nilai acak.

^2: Menghitung kuadrat dari setiap elemen dalam array.

lot: Menggambar titik-titik fungsi.

---

2

---

Menggambar Fungsi Linear

$$f(x) = 2x + 1$$

**Penyelesaian :**

---

```
>clc; // clear screen
>window(0, 1024, 0, 1024); // use all of the window
>setplot(-10, 10, -10, 30); // set plot coordinates
>hold on; // start overwrite mode
>// Definisikan jumlah titik
>n = 200;
```

```
>x = linspace(-10, 10, n); // nilai x dari -10 hingga 10
>y = 2*x + 1; // hitung y untuk f(x) = 2x + 1
>colors=rgb(random(n),random(n),random(n)); // get random colors
>// Gambar kurva
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
```

```
Matrices must fit for plot!
Got 1x1 and 1000x2
Try "trace errors" to inspect local variables after errors.
plot:
    res = _plot(x,y);
```

```
>hold off; // end overwrite mode
>insimg; // insert to notebook
```

---

```
>reset;
```

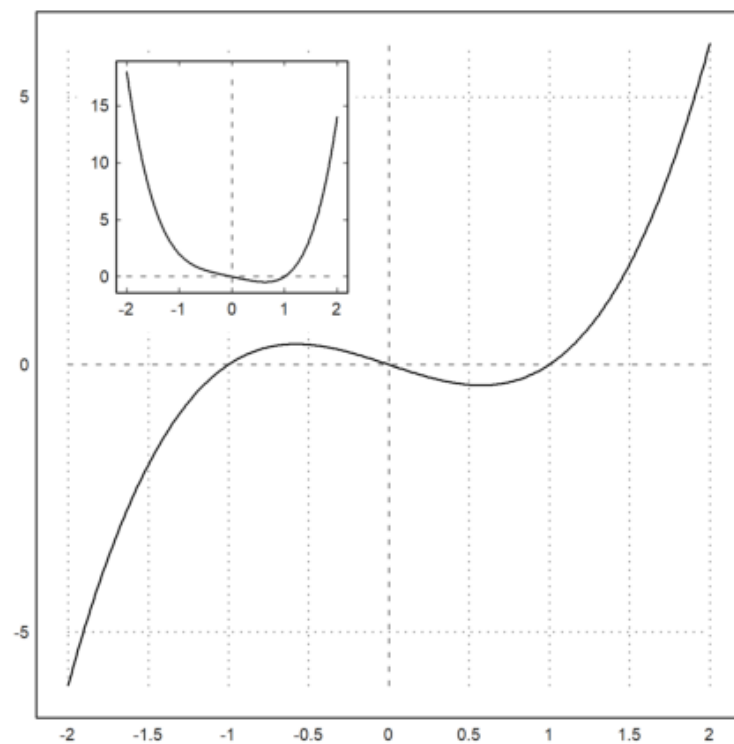
Grafik perlu ditahan, karena perintah plot() akan menghapus jendela plot.

Untuk menghapus semua yang kami lakukan, kami menggunakan reset().

Untuk menampilkan gambar hasil plot di layar notebook, perintah plot2d() dapat diakhiri dengan titik dua (:). Cara lain adalah perintah plot2d() diakhiri dengan titik koma (;), kemudian menggunakan perintah insimg() untuk menampilkan gambar hasil plot.

Untuk contoh lain, kami menggambar plot sebagai sisipan di plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan bahwa kami menyimpan dan memulihkan jendela penuh, dan menahan plot saat ini saat kami memplot inset.

```
>plot2d("x^3-x");
>xw=200; yw=100; ww=300; hw=300;
>ow=window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^4-x",grid=6):
```



```
>hold off;
>window(ow);
```

Plot dengan banyak angka dicapai dengan cara yang sama. Ada fungsi `figure()` utilitas untuk ini.

## Catatan Baru

### 1. `plot2d("x^3-x");`

- Fungsi ini: Menggambar grafik dari fungsi polinomial  $f(x) = x^3 - x$ .
- Hasil: Menampilkan grafik utama di jendela plot.

2. `xw=200; yw=100; ww=300; hw=300;`

- Variabel: Mendefinisikan posisi dan ukuran untuk jendela inset.
- `'xw'` dan `'yw'` adalah koordinat (x, y) dari sudut kiri atas jendela

inset.

- `'ww'` adalah lebar jendela inset.
- `'hw'` adalah tinggi jendela inset.

3. `'ow=window();'`

- Fungsi ini: Menyimpan referensi ke jendela plot saat ini (jendela

utama).

- Tujuan: Memungkinkan untuk kembali ke jendela utama setelah

menggambar inset.

4. `'window(xw,yw,xw+ww,yw+hw);'`

- Fungsi ini: Mengatur jendela baru untuk menggambar inset.
- Parameter: Menggunakan koordinat dan ukuran yang sudah

didefinisikan untuk menentukan posisi jendela inset.

5. `'hold on;'`

- Fungsi ini: Memungkinkan untuk menambahkan beberapa plot dalam

satu jendela tanpa menghapus plot yang sudah ada.

- Konteks: Digunakan sebelum menggambar fungsi inset agar grafik

sebelumnya tidak hilang.

6. `'barclear(xw-50,yw-10,ww+60,ww+60);'`

- Fungsi ini: Membersihkan area plot pada jendela inset sebelum

menggambar fungsi baru.

- Parameter: Mengatur area yang akan dibersihkan, memberikan sedikit

margin di sekitar jendela inset.

7. `'plot2d("x^4-x",grid=6):'`

- Fungsi ini: Menggambar grafik fungsi  $(f(x) = x^4 - x)$  dalam

jendela inset.

- Parameter `'grid=6'`: Menambahkan grid ke grafik untuk meningkatkan

keterbacaan.

- Titik dua (``:``): Menggunakan titik dua untuk menampilkan grafik

langsung di layar notebook.

8. `'hold off;'`

- Fungsi ini: Mengakhiri mode penahanan, sehingga tidak ada grafik tambahan yang akan ditambahkan ke jendela saat ini.

9. `'window(ow);'`

- Fungsi ini: Mengembalikan tampilan ke jendela plot utama yang disimpan sebelumnya.

- Tujuan: Memastikan pengguna dapat melihat grafik utama kembali setelah menggambar inset.

Ringkasan

Secara keseluruhan, kode ini menggambarkan dua fungsi berbeda,  $(x^3 - x)$  di jendela utama dan  $(x^4 - x)$  di jendela inset. Proses ini melibatkan pengaturan jendela baru untuk menampilkan grafik inset tanpa mengganggu grafik utama, serta menambahkan grid untuk membantu visualisasi. Jika ada yang ingin ditanyakan lebih lanjut, silakan!

## contoh penyelesaian masalah menggambar kurva/plot 2D

1

Menggambar Fungsi Trigonometri

$$f(x) = \sin(x)$$

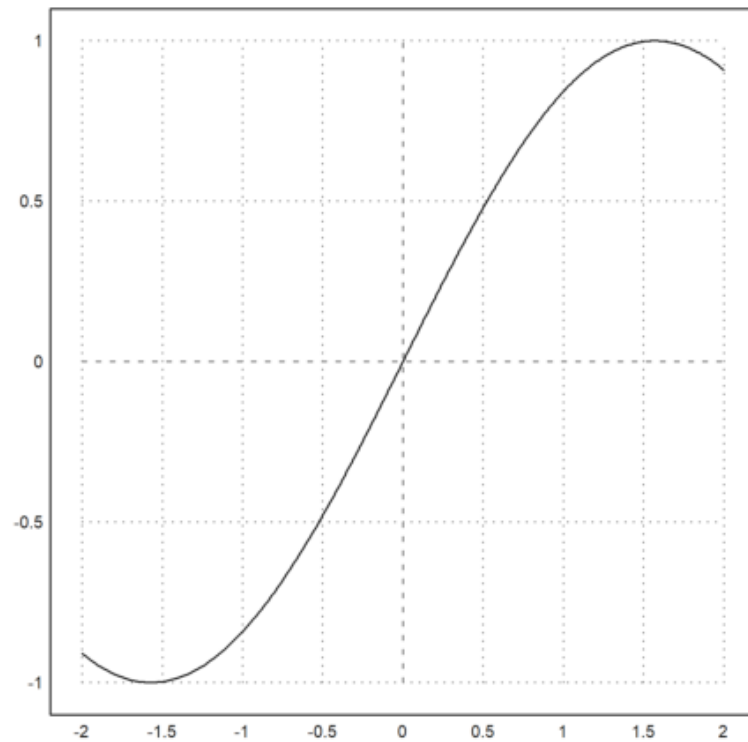
dengan Inset Plot

**Penyelesaian :**

```
>reset; // menghapus semua plot sebelumnya
```

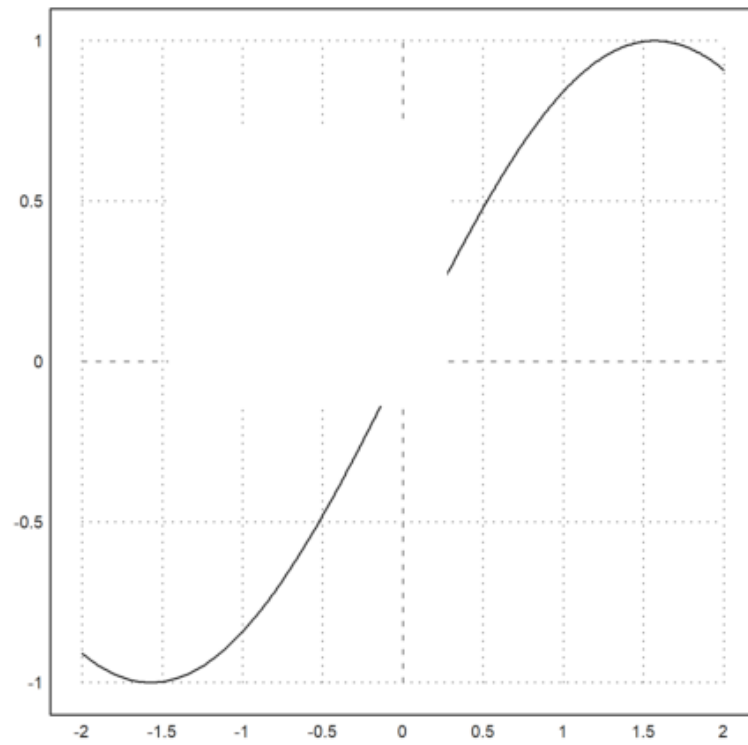
```
>plot2d("sin(x)"): // menggambar grafik fungsi utama
```



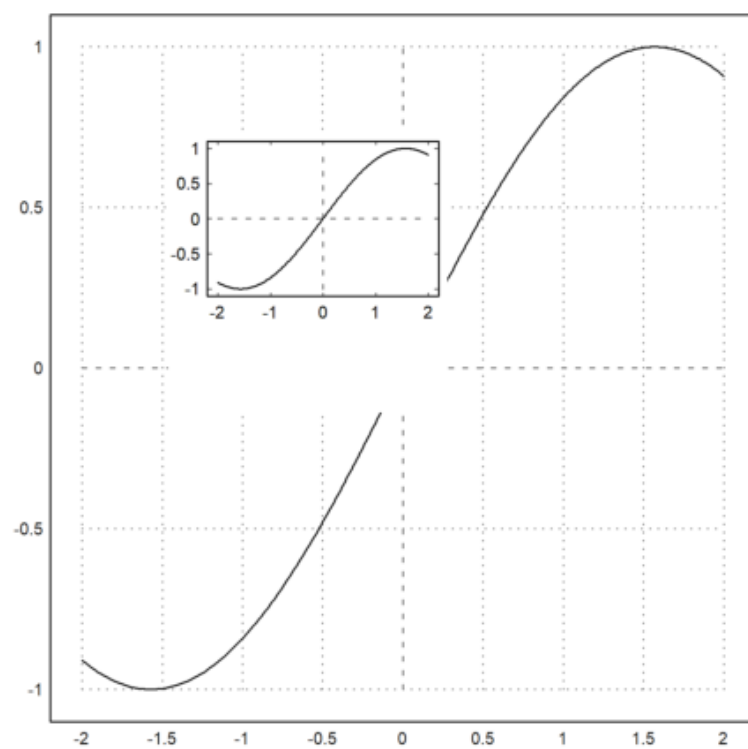


```
>xw = 300; yw = 200; ww = 300; hw = 200; // mendefinisikan posisi dan ukuran jendela inset  
>ow = window(); // menyimpan jendela penuh saat ini
```

```
>window(xw, yw, xw + ww, yw + hw); // mengatur jendela baru untuk inset plot  
>hold on; // menahan plot  
>barclear(xw - 50, yw - 10, ww + 60, ww + 60): // membersihkan area plot jika diperlukan
```



```
>plot2d("sin(x)", grid = 6): // menggambar fungsi inset
```



```
>hold off; // mengakhiri penahanan plot
>window(ow); // mengembalikan ke jendela penuh
```

Penjelasan:

- Grafik utama menampilkan fungsi sinus.
- Inset digunakan untuk menunjukkan lebih detail pada fungsi sinus dalam jendela terpisah.
- grid = 6 menambahkan grid ke grafik untuk meningkatkan keterbacaan.

---

2

---

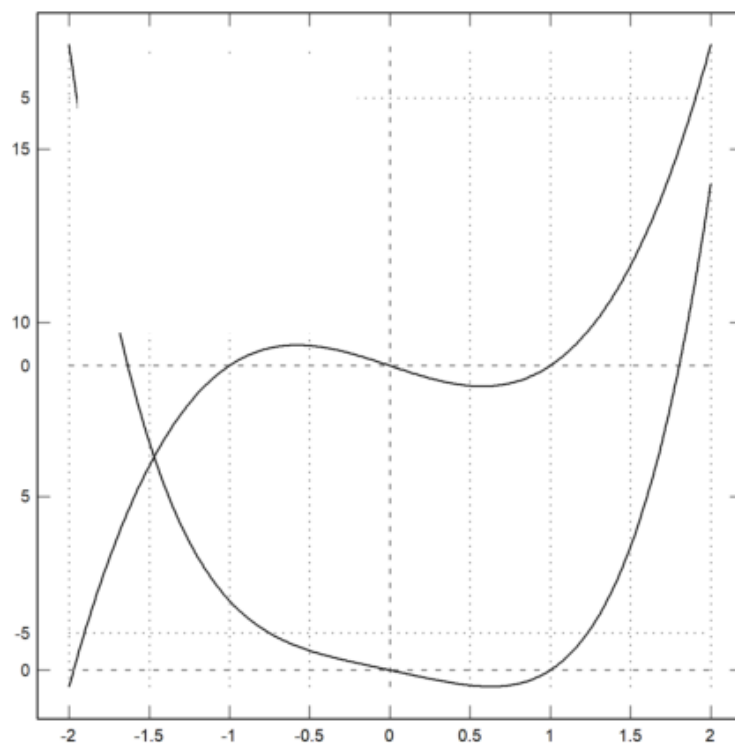
Menggambar Inset Plot untuk Fungsi

$$f(x) = x^4 - x$$

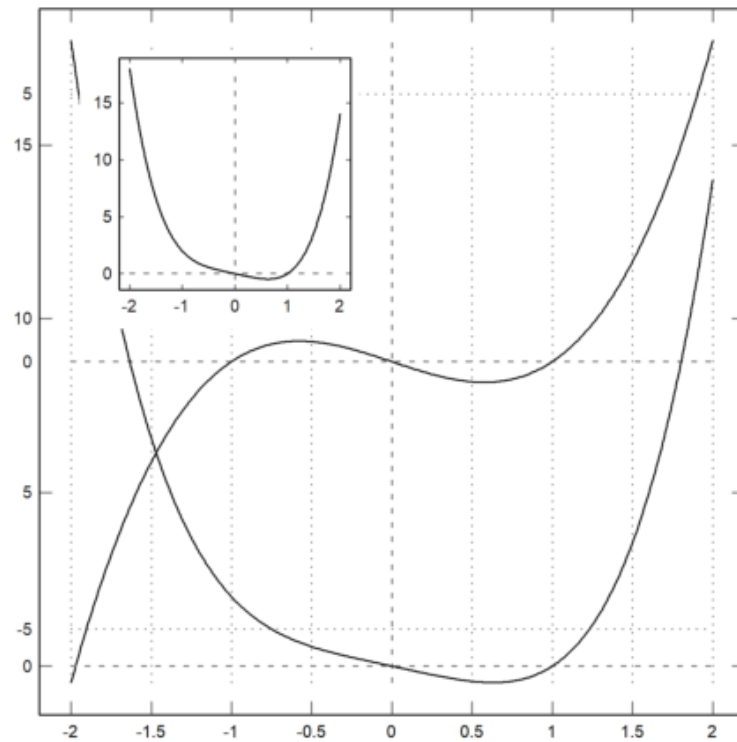
**Penyelesaian :**

---

```
>xw = 200; yw = 100; ww = 300; hw = 300; // mendefinisikan posisi dan ukuran jendela inset
>ow = window(); // menyimpan jendela penuh saat ini
>window(xw, yw, xw + ww, yw + hw); // mengatur jendela baru untuk inset plot
>hold on; // menahan plot
>barclear(xw - 50, yw - 10, ww + 60, ww + 60): // membersihkan area plot jika diperlukan
```



```
>plot2d("x^4 - x", grid = 6): // menggambar fungsi inset
```



```
>hold off; // mengakhiri penahanan plot
>window(ow); // mengembalikan ke jendela penuh
```

Penjelasan:

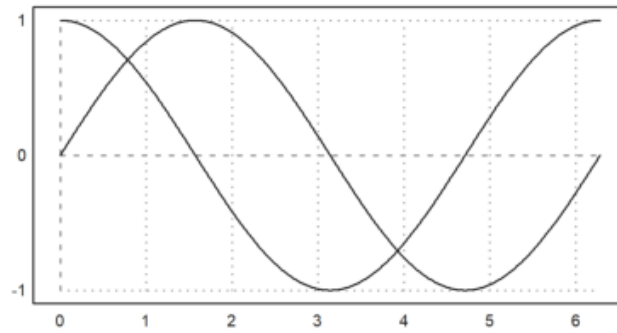
- window() menyimpan jendela penuh untuk memulihkannya nanti.
- barclear() digunakan untuk memastikan area plot bersih sebelum menggambar fungsi inset.
- plot2d("x^4 - x", grid = 6) menggambar fungsi polinomial  $x^4 - x$  di dalam jendela inset.

## Aspek Plot

Plot default menggunakan jendela plot persegi. Anda dapat mengubah ini dengan fungsi `aspect()`. Jangan lupa untuk mengatur ulang aspek nanti. Anda juga dapat mengubah default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran jendela grafis saat ini.

Tetapi Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area plot saat ini diubah, dan jendela diatur sehingga label memiliki cukup ruang.

```
>aspect(2); // rasio panjang dan lebar 2:1
>plot2d(["sin(x)", "cos(x)"], 0, 2pi):
```



```
>aspect();
>reset;
```

Fungsi reset() mengembalikan plot default termasuk rasio aspek

---

## Catatan Baru

---

1. aspect(2);

Fungsi ini: Mengatur rasio aspek grafik.

Dalam hal ini, rasio panjang dan lebar diatur menjadi 2:1. Ini berarti grafik akan lebih lebar dibandingkan tingginya.

2. plot2d(["sin(x)", "cos(x)"], 0, 2\*%pi);

- Fungsi ini: Menggambar grafik untuk fungsi sinus dan kosinus dari 0 hingga 2pi.

- format: Menggunakan array untuk menggambar dua fungsi sekaligus.

3. spect();

-Fungsi ini: Mengembalikan pengaturan rasio aspek ke default.

4. reset;

-Fungsi ini: Menghapus semua pengaturan plot yang ada, sehingga siap untuk plot baru.

## contoh penyelesaian masalah menggambar kurva/plot 2D

---

1

---

Menggambar Grafik Fungsi Sinus dengan Rasio Aspek

1 : 1

**Penyelesaian :**

---

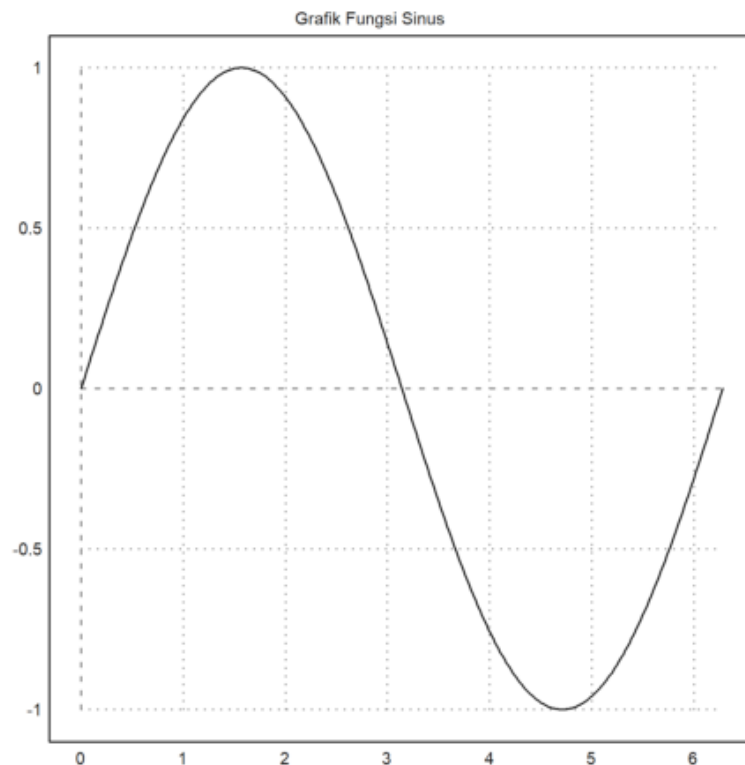
```
>aspect(1); // rasio panjang dan lebar 1:1
>plot2d("sin(x)", 0, 2*pi); // menggambar grafik fungsi sinus
>title("Grafik Fungsi Sinus"); // judul
>xl("x"); // label sumbu x
```

```
Function xl not found.
Try list ... to find functions!
Error in:
xl("x"); // label sumbu x ...
  ^
```

```
>yl("sin(x)"); // label sumbu y
```

```
Function yl not found.
Try list ... to find functions!
Error in:
yl("sin(x)"); // label sumbu y ...
  ^
```

```
>aspect(); // mengembalikan rasio aspek ke default
>insimg; // menampilkan gambar hasil plot
```



#### Penjelasan:

rafik sinus ditampilkan dengan rasio aspek 1:1, sehingga sumbu x dan y memiliki panjang yang sama.

## Menggambar Grafik Fungsi Kuadrat

$$f(x) = x^2$$

dengan Rasio Aspek 4:3

**Penyelesaian :**

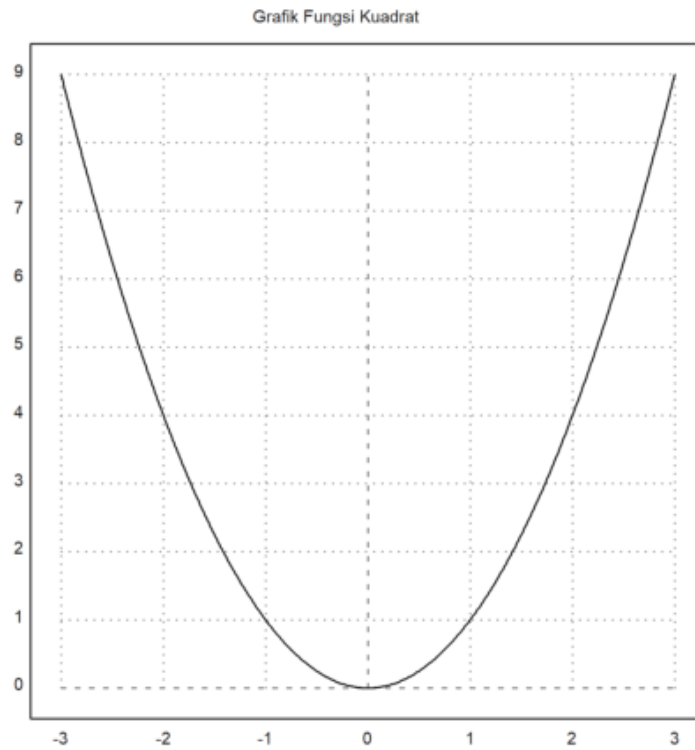
```
>aspect(4/3); // rasio panjang dan lebar 4:3
>plot2d("x^2", -3, 3); // menggambar grafik fungsi kuadrat
>title("Grafik Fungsi Kuadrat"); // judul
>xl("x"); // label sumbu x
```

```
Function xl not found.
Try list ... to find functions!
Error in:
xl("x"); // label sumbu x ...
  ^
```

```
>yl("x^2"); // label sumbu y
```

```
Function yl not found.
Try list ... to find functions!
Error in:
yl("x^2"); // label sumbu y ...
  ^
```

```
>aspect(); // mengembalikan rasio aspek ke default
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

rafik fungsi kuadrat ditampilkan dengan rasio aspek 4:3, yang memberikan tampilan lebih lebar.

## Plot 2D di Euler

EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi `plot2d`. Fungsi ini dapat memplot fungsi dan data.

Dimungkinkan untuk membuat plot di Maxima menggunakan Gnuplot atau dengan Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

- ekspresi
- fungsi, variabel, atau kurva parameter,
- vektor nilai x-y,
- awan titik di pesawat,
- kurva implisit dengan level atau wilayah level.
- Fungsi kompleks

Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang dan plot berbayang.

## Catatan Baru

### contoh penyelesaian masalah menggambar kurva/plot 2D



## Menggambar Grafik Fungsi Linier

$$f(x) = 2x + 1$$

### Penyelesaian :

---

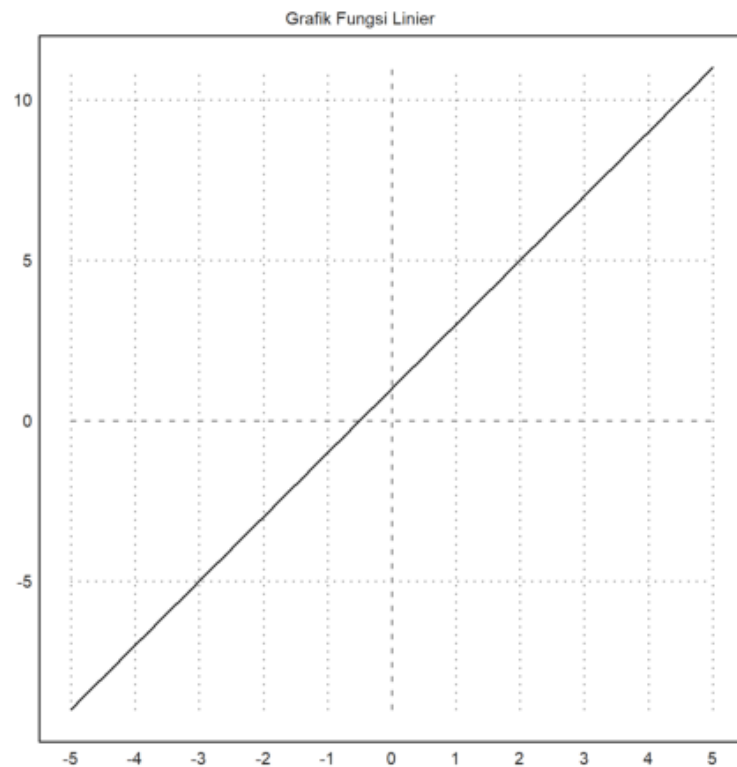
```
> reset; // menghapus semua plot sebelumnya
> plot2d("2*x + 1", -5, 5); // menggambar grafik fungsi linier
> title("Grafik Fungsi Linier"); // memberikan judul
> xl("x"); // label sumbu x
```

```
Function xl not found.
Try list ... to find functions!
Error in:
xl("x"); // label sumbu x ...
      ^
```

```
> yl("f(x) = 2x + 1"); // label sumbu y
```

```
Function yl not found.
Try list ... to find functions!
Error in:
yl("f(x) = 2x + 1"); // label sumbu y ...
      ^
```

```
> insimg; // menampilkan gambar hasil plot
```



Penjelasan:

fungsi linier digambar dari -5 hingga 5 memberikan visualisasi yang jelas tentang hubungan linier antara  $x$  dan  $f(x)$ .

---

2

---

Menggambar Grafik Fungsi Kuadrat

$$f(x) = x^2 - 4$$

**Penyelesaian:**

---

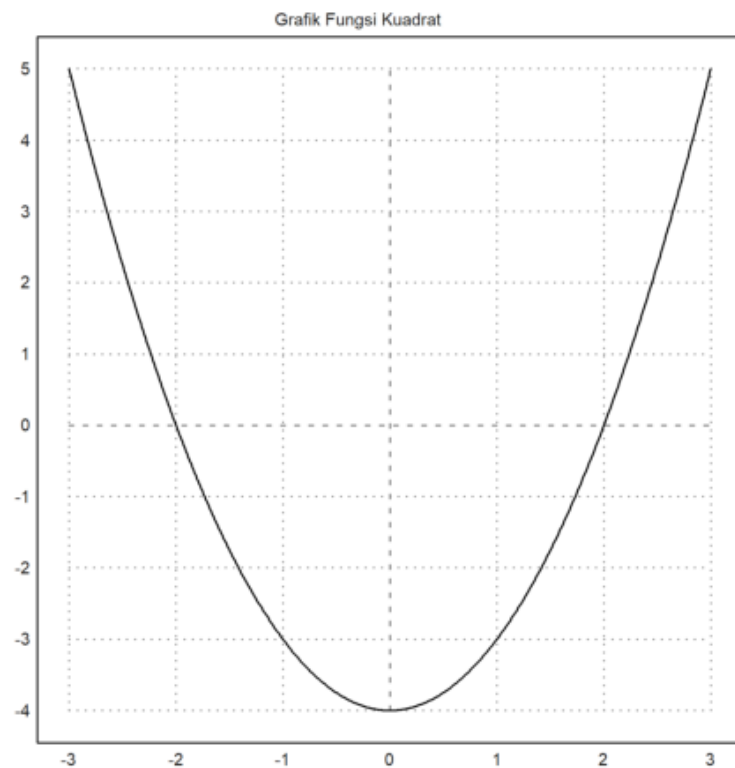
```
>reset; // menghapus semua plot sebelumnya
>plot2d("x^2 - 4", -3, 3); // menggambar grafik fungsi kuadrat
>title("Grafik Fungsi Kuadrat"); // memberikan judul
>xl("x"); // label sumbu x
```

```
Function xl not found.
Try list ... to find functions!
Error in:
xl("x"); // label sumbu x ...
^
```

```
>yl("f(x) = x^2 - 4"); // label sumbu y
```

```
Function yl not found.  
Try list ... to find functions!  
Error in:  
yl("f(x) = x^2 - 4"); // label sumbu y ...  
      ^
```

```
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

grafik fungsi kuadrat ditampilkan dari -3 hingga 3, memperlihatkan parabola yang terbuka ke atas dengan titik potong di sumbu y.

---

## Plot Ekspresi atau Variabel

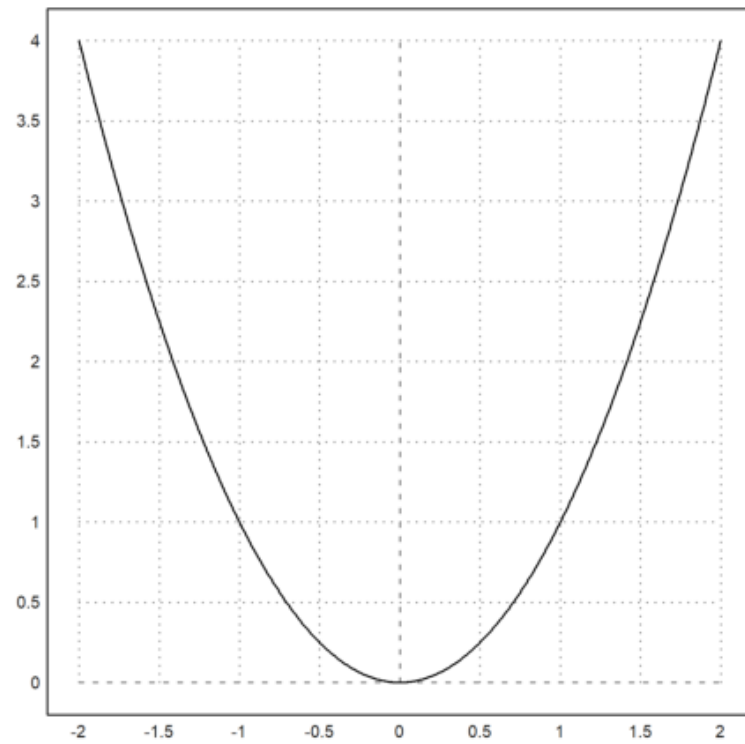
---

Ekspresi tunggal dalam "x" (mis. " $4x^2$ ") atau nama fungsi (mis. "f") menghasilkan grafik fungsi.

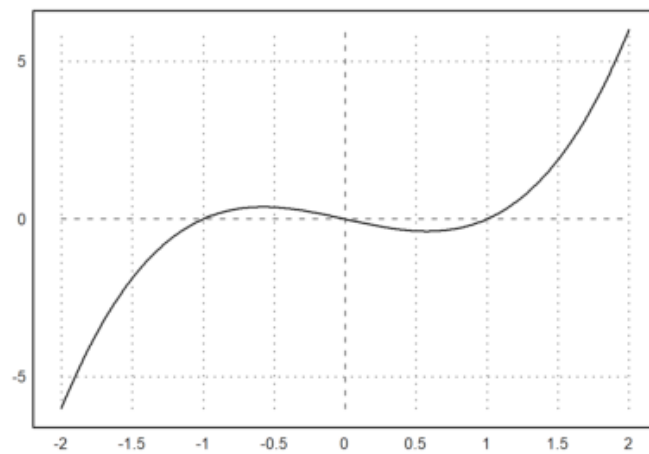
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

Catatan: Jika Anda mengakhiri baris perintah dengan titik dua ":", plot akan dimasukkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

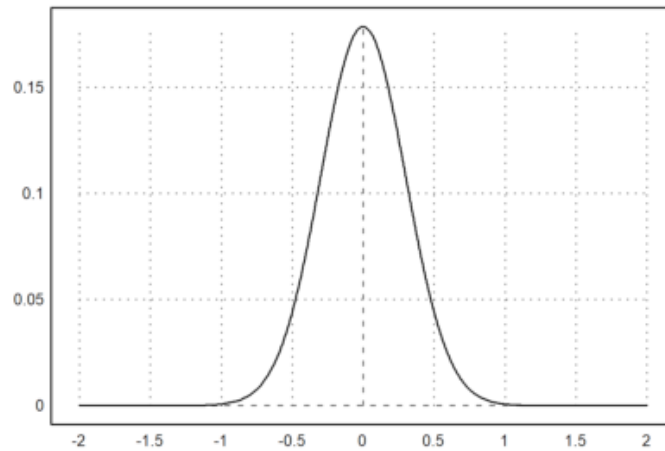
```
>plot2d("x^2") :
```



```
>aspect(1.5); plot2d("x^3-x") :
```



```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```

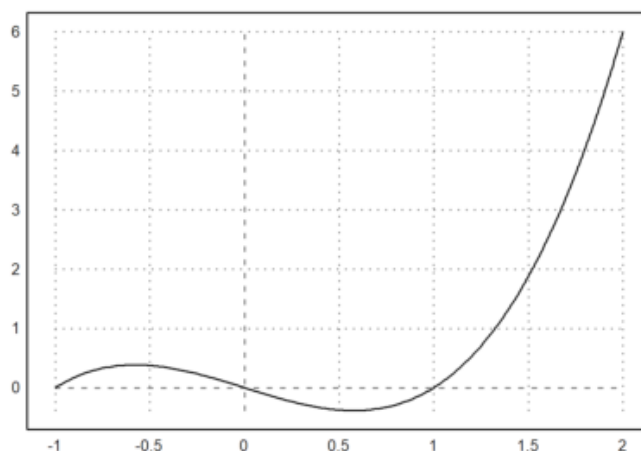


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

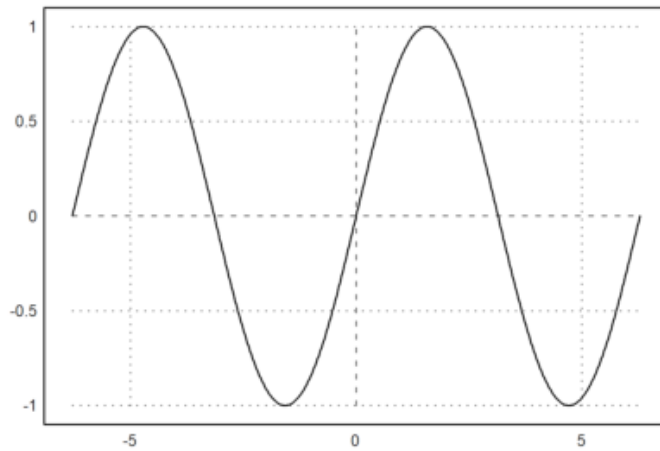
Rentang plot diatur dengan parameter yang ditetapkan sebagai berikut

- a,b: rentang x (default -2,2)
- c,d: rentang y (default: skala dengan nilai)
- r: alternatifnya radius di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

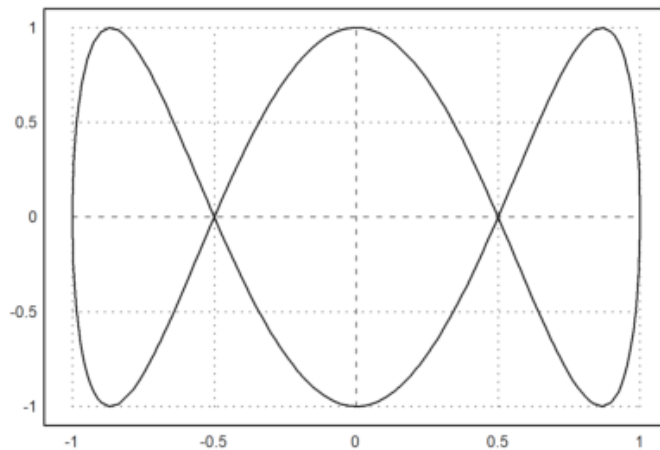
```
>plot2d("x^3-x",-1,2):
```



```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]
```



```
>plot2d("cos(x)", "sin(3*x)", xmin=0, xmax=2pi):
```



Alternatif untuk titik dua adalah perintah `insimg(baris)`, yang menyisipkan plot yang menempati sejumlah baris teks tertentu.

Dalam opsi, plot dapat diatur untuk muncul

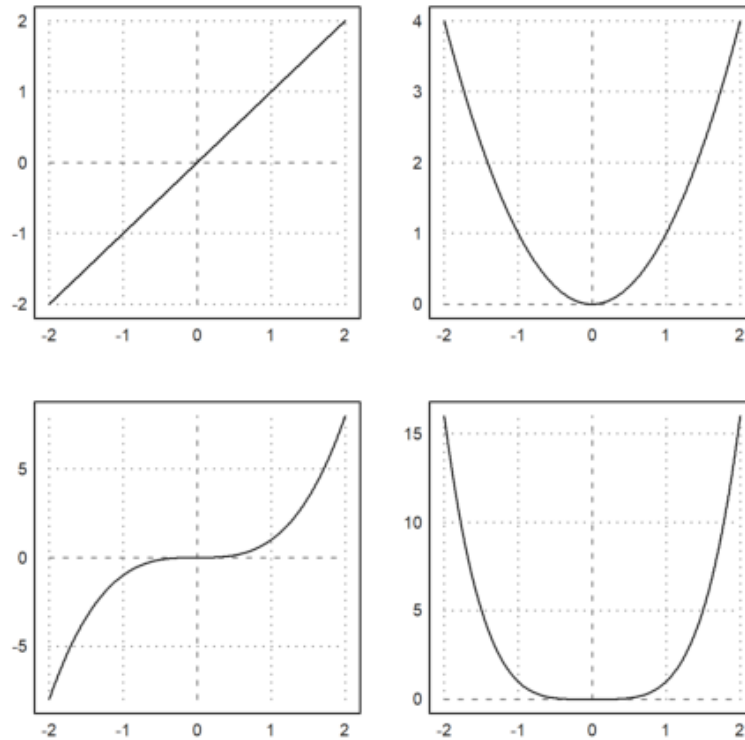
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Bagaimanapun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.

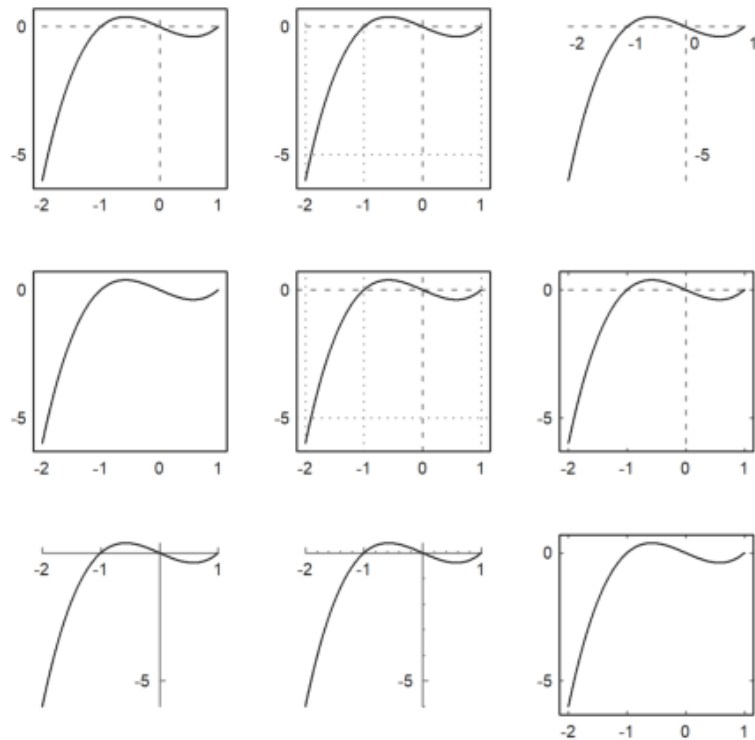
Untuk membagi jendela menjadi beberapa plot, gunakan perintah `figure()`. Dalam contoh, kami memplot  $x^1$  hingga  $x^4$  menjadi 4 bagian jendela. `figure(0)` mengatur ulang jendela default.

```
>reset;
>figure(2,2); ...
>for n=1 to 4; figure(n); plot2d("x^"+n); end; ...
>figure(0):
```



Di `plot2d()`, ada gaya alternatif yang tersedia dengan `grid=x`. Untuk gambaran umum, kami menunjukkan berbagai gaya kisi dalam satu gambar (lihat di bawah untuk perintah `figure()`). Gaya `kisi=0` tidak disertakan. Ini menunjukkan tidak ada grid dan tidak ada bingkai.

```
>figure(3,3); ...
>for k=1:9; figure(k); plot2d("x^3-x",-2,1,grid=k); end; ...
>figure(0):
```

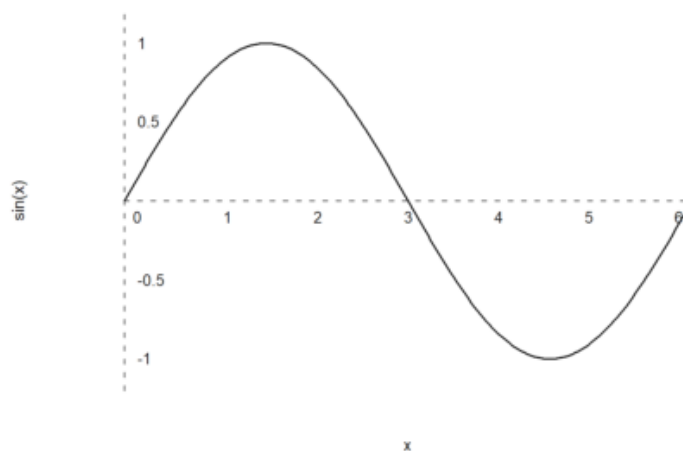


Jika argumen ke `plot2d()` adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang  $x$  dan  $y$  untuk plot.

Atau,  $a$ ,  $b$ ,  $c$ ,  $d$  dapat ditentukan sebagai parameter yang ditetapkan sebagai  $a=...$  dll.

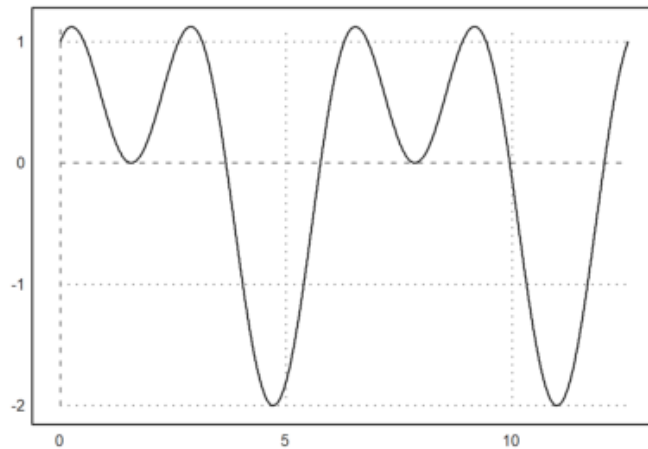
Dalam contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu  $y$ .

```
>aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)") :
```



```
>plot2d("sin(x)+cos(2*x)",0,4pi) :
```



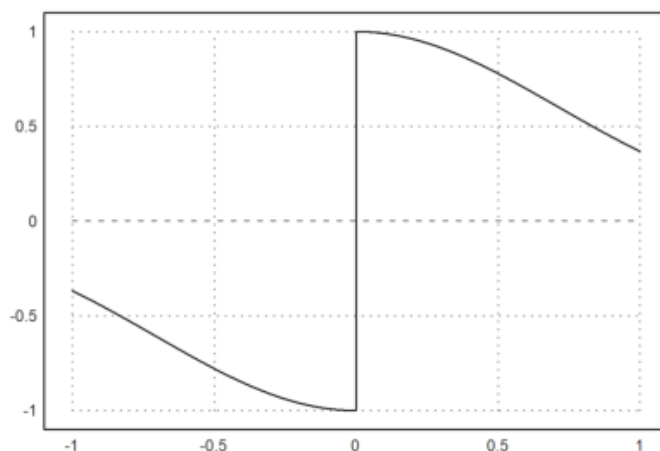


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "gambar". Mereka juga digunakan oleh ekspor HTML.

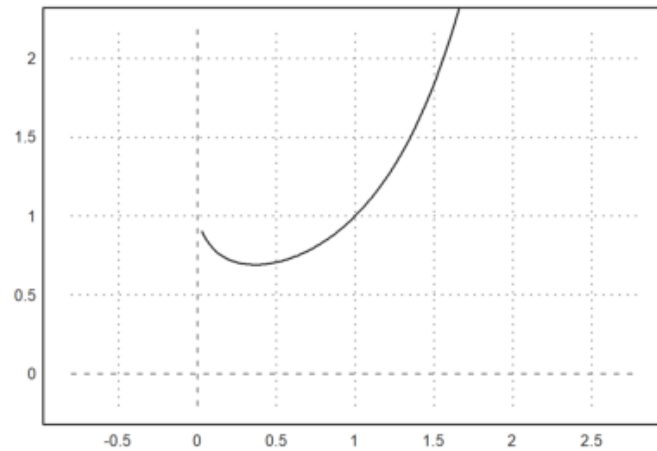
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengeksport grafik saat ini dengan fungsi di menu File.

Fungsi atau ekspresi dalam `plot2d` dievaluasi secara adaptif. Untuk kecepatan lebih, matikan plot adaptif dengan `<adaptive` dan tentukan jumlah subinterval dengan `n=...`. Ini hanya diperlukan dalam kasus yang jarang terjadi.

```
>plot2d("sign(x)*exp(-x^2)",-1,1,<adaptive,n=10000):
```

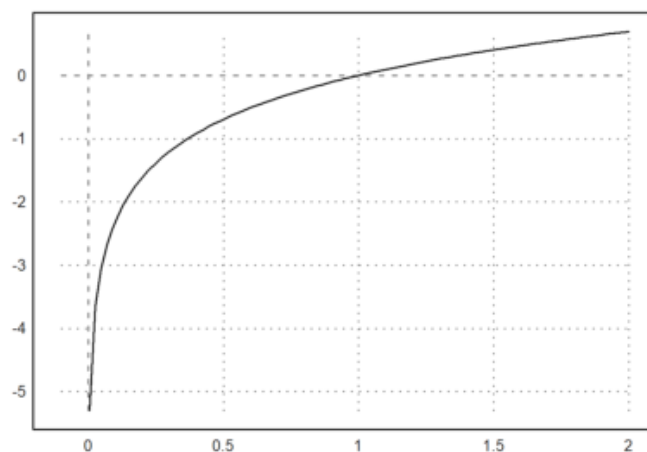


```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



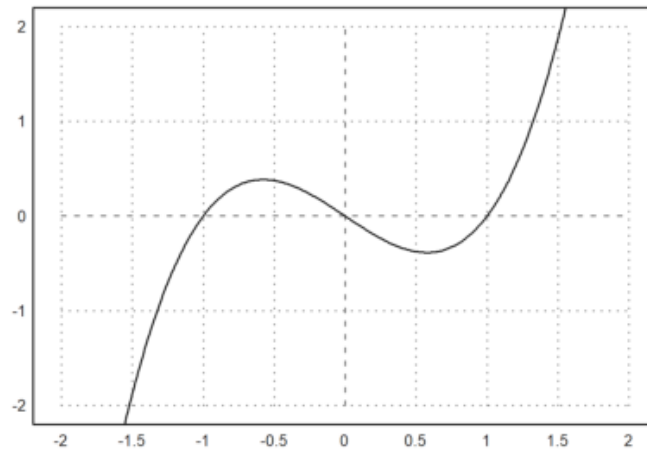
Perhatikan bahwa  $x^x$  tidak didefinisikan untuk  $x \leq 0$ . Fungsi `plot2d` menangkap kesalahan ini, dan mulai merencanakan segera setelah fungsi didefinisikan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN keluar dari jangkauan definisinya.

```
>plot2d("log(x)",-0.1,2):
```

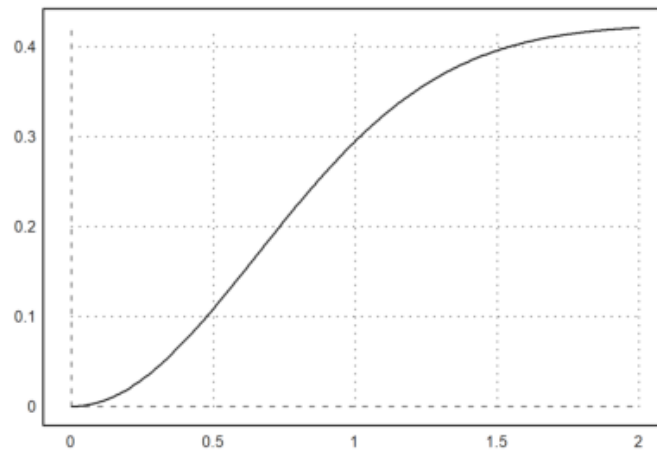


Parameter `square=true` (atau `>square`) memilih y-range secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

```
>plot2d("x^3-x",>square):
```

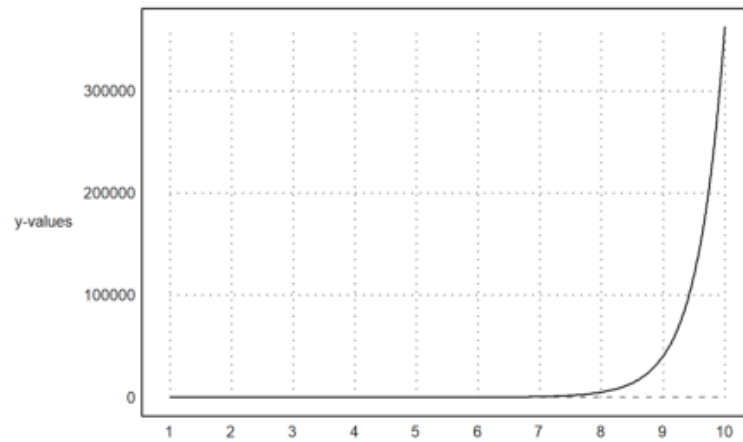


```
>plot2d(''integrate("sin(x)*exp(-x^2)","0,x)','',0,2): // plot integral
```



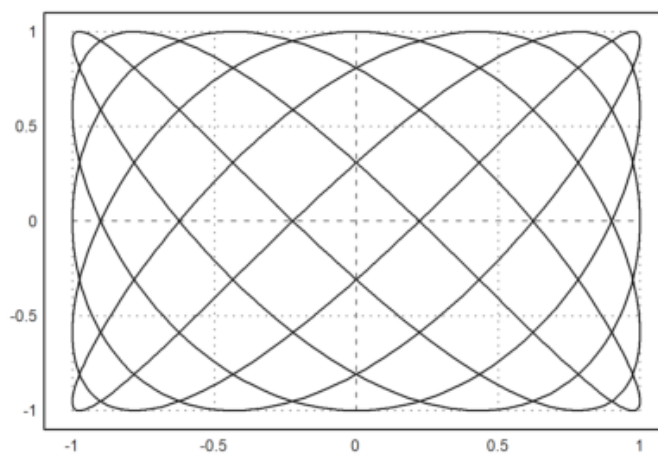
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil `shrinkwindow()` dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di `plot2d()`.

```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```

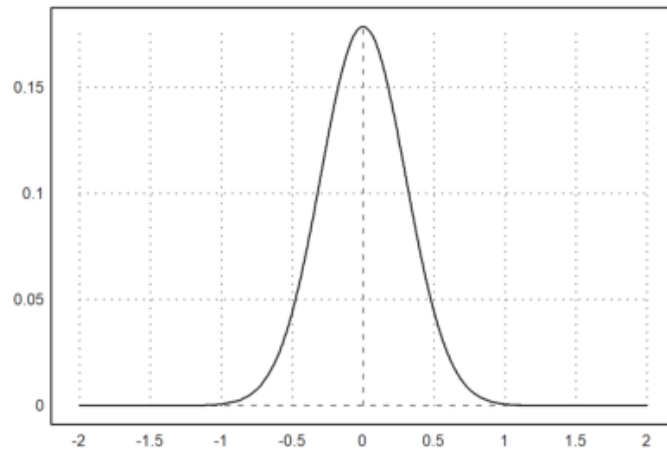


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

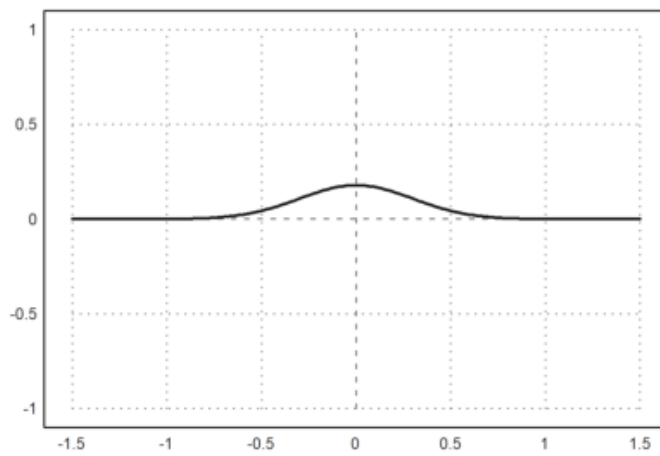
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



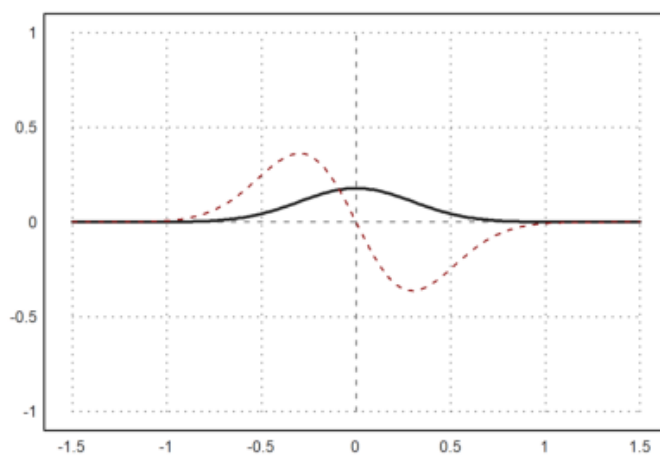
```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



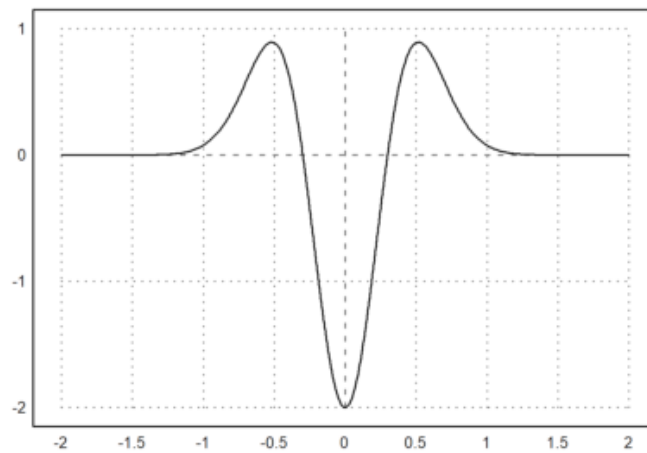
```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



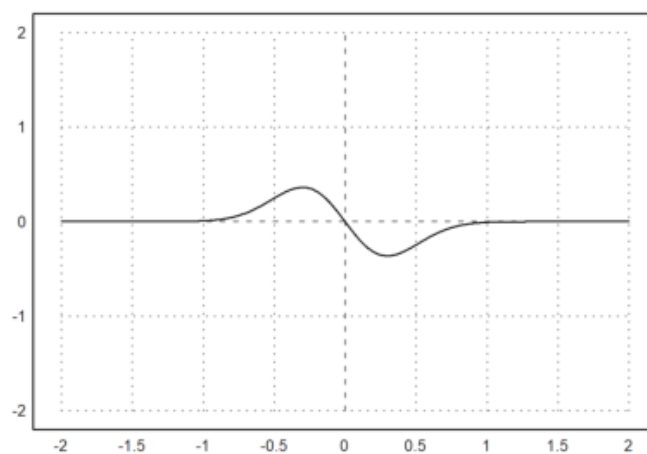
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



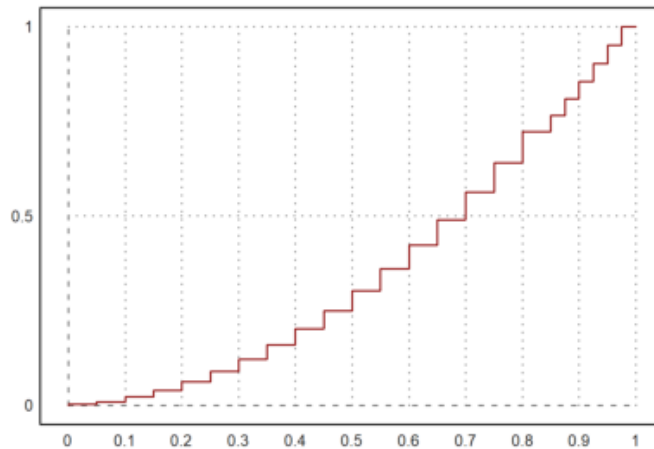
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



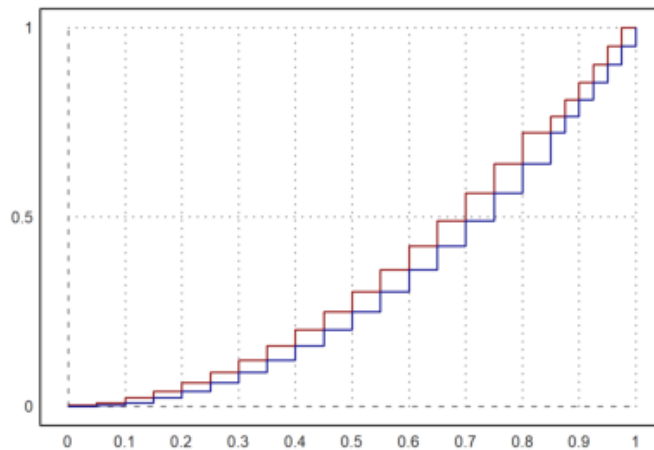
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```



```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
```



```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



## Catatan Baru

EMT menyediakan fungsi `plot2d` yang memungkinkan pengguna untuk menggambar grafik dari ekspresi atau variabel matematis. Dengan `plot2d`, kita dapat:

- Menggambarkan ekspresi dalam variabel  $x$ , seperti  $4x^2$ .
- Mengatur rentang untuk sumbu  $x$  dan  $y$  secara manual.
- Menggunakan berbagai opsi untuk menyesuaikan gaya plot, termasuk grid, label, dan aspek rasio.

Fungsi ini mendukung plot yang lebih kompleks, seperti fungsi integral dan diferensial, serta penggunaan ekspresi simbolik.

## contoh penyelesaian masalah menggambar kurva/plot 2D

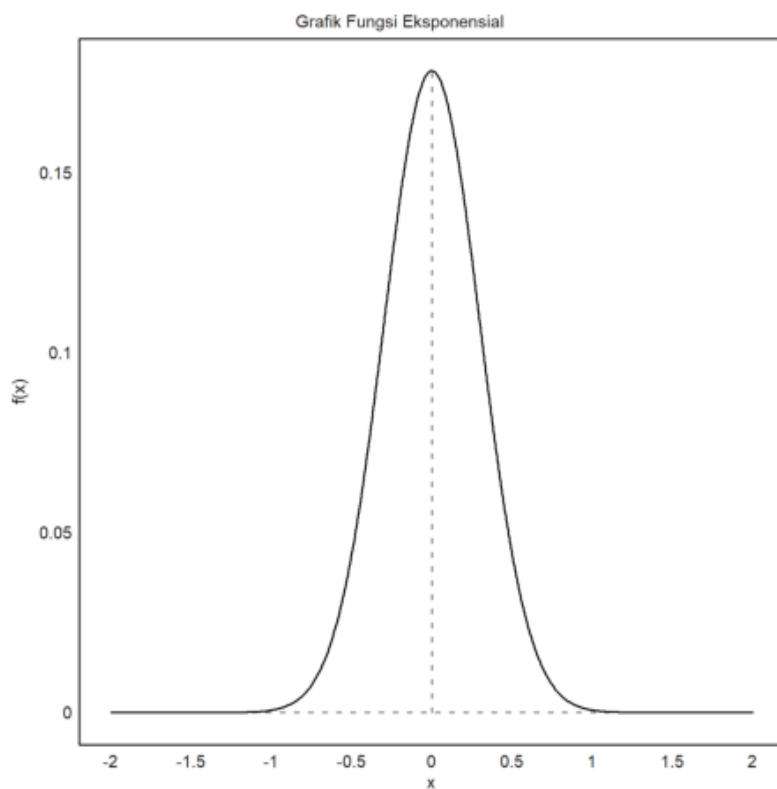
## Menggambar Grafik Fungsi Eksponensial

$$f(x) = (e^{-x})^2$$

**Penyelesaian :**

---

```
>reset; // menghapus semua plot sebelumnya  
>a := 5.6; // mendefinisikan parameter  
>plot2d("exp(-a*x^2)/a", -2, 2, grid=1, xl="x", yl="f(x)"); // menggambar grafik fungsi ek  
>title("Grafik Fungsi Eksponensial"); // memberikan judul  
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

ungsi eksponensial digambar pada interval  
2 hingga

2. Grid ditambahkan untuk meningkatkan keterbacaan.



## Menggambar Grafik Fungsi Sinus

$$f(x) = \sin(x)$$

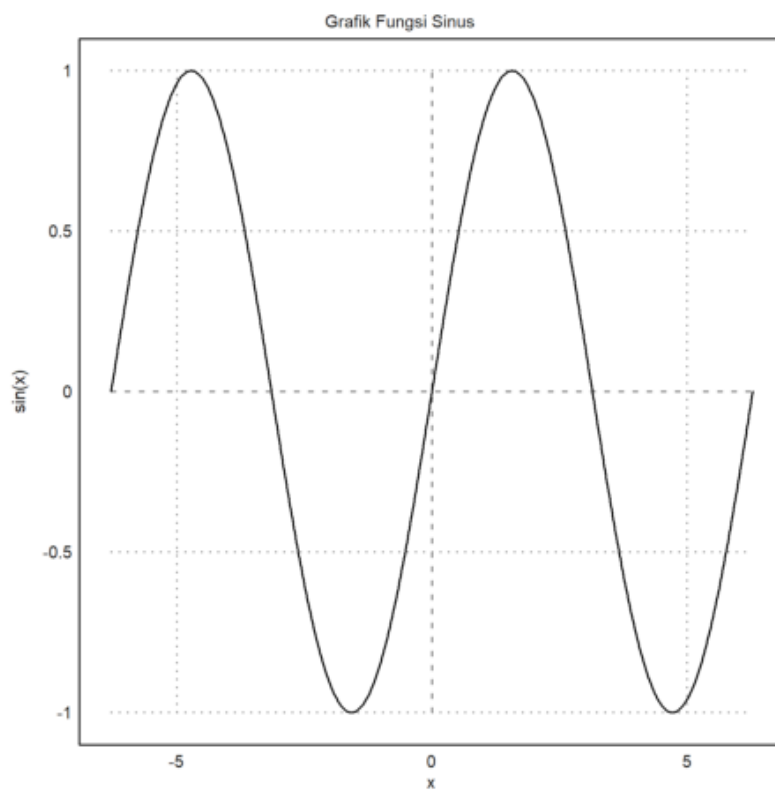
dan Kosinus

$$f(x) = \cos(x)$$

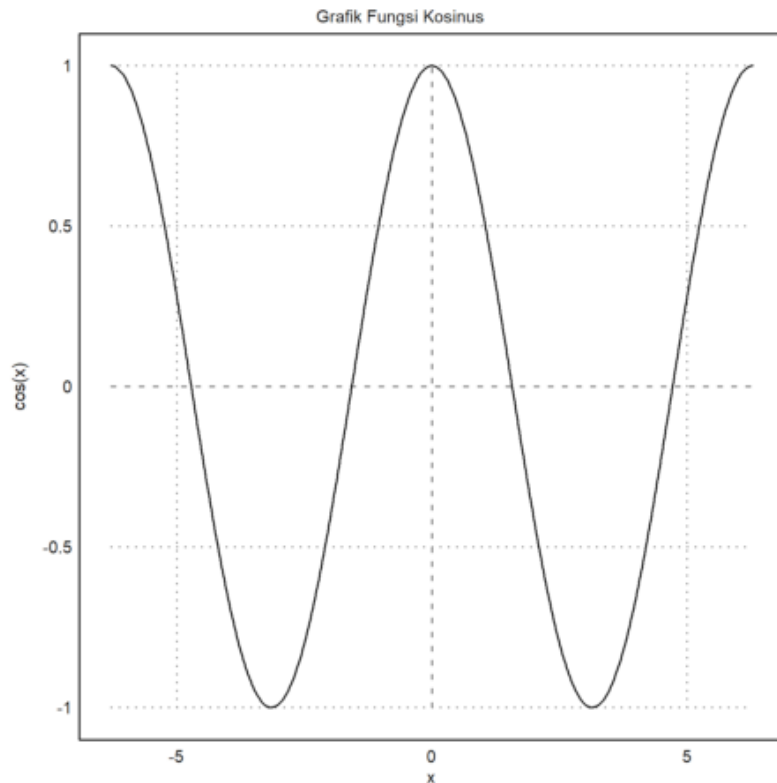
**Penyelesaian:**

---

```
>reset; // menghapus semua plot sebelumnya  
>plot2d("sin(x)", -2*pi, 2*pi, grid=2, xl="x", yl="sin(x)"); // menggambar grafik fungsi  
>title("Grafik Fungsi Sinus"); // memberikan judul  
>insimg; // menampilkan gambar hasil plot
```



```
>plot2d("cos(x)", -2*pi, 2*pi, grid=2, xl="x", yl="cos(x)"); // menggambar grafik fungsi  
>title("Grafik Fungsi Kosinus"); // memberikan judul  
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

Grafik fungsi sinus dan kosinus ditampilkan pada interval  $-2\pi$  hingga  $2\pi$ , dengan grid untuk membantu visualisasi fungsi yang berosilasi.

---

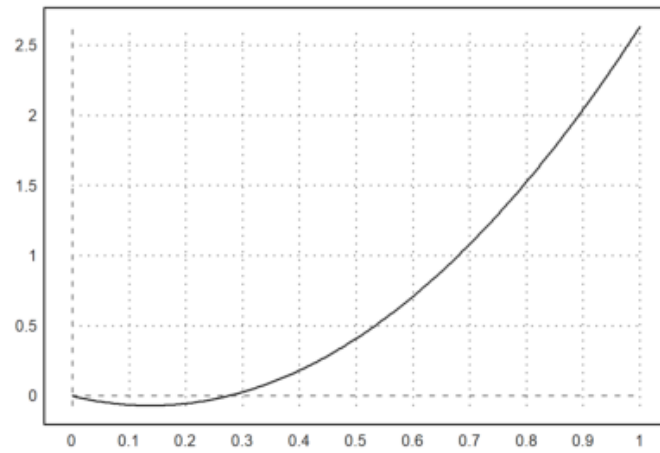
## Fungsi dalam satu Parameter

---

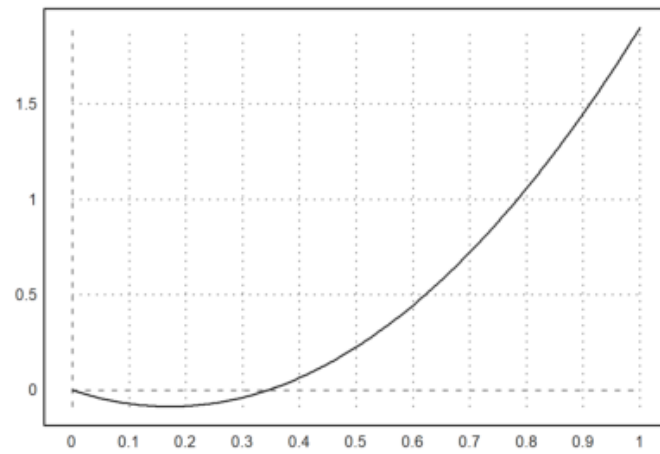
Fungsi plot yang paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.

Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda dapat meneruskan parameter tambahan (selain  $x$ ) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

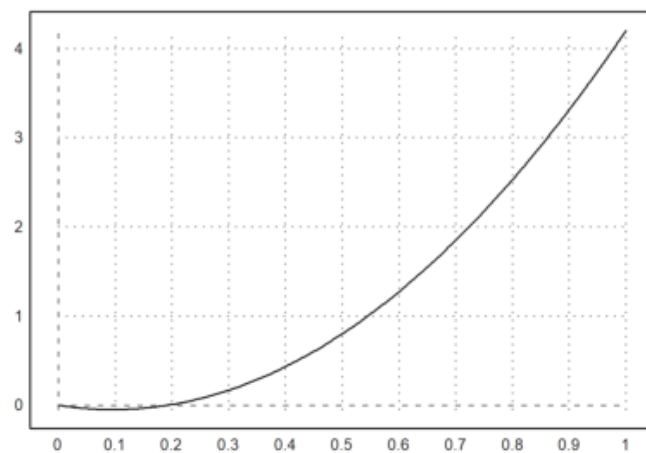
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



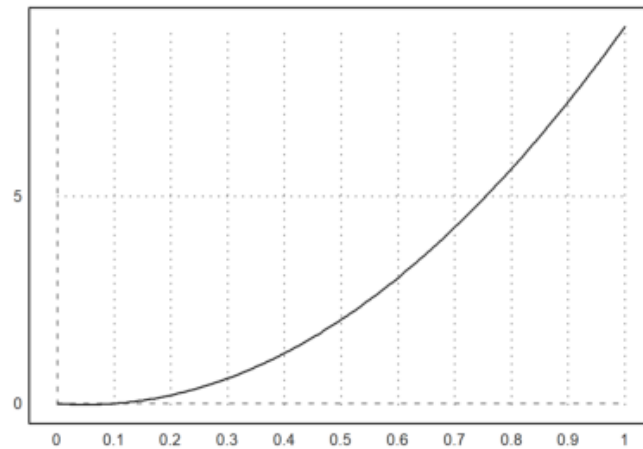
```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



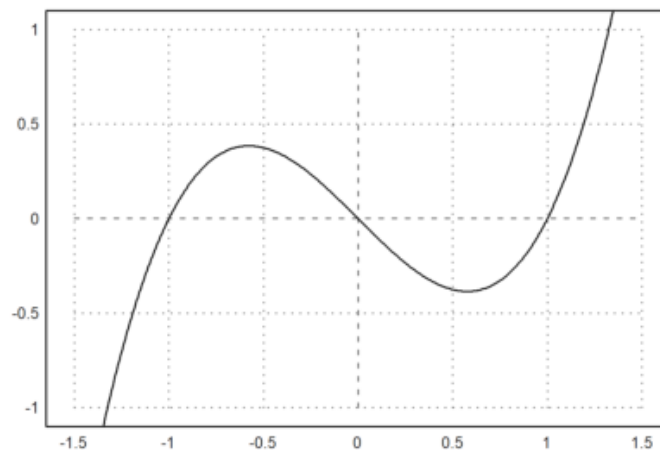
```
>plot2d({"f",0.2},0,1): // plot with a=0.2
```



```
>plot2d({{"f(x,b)",b=0.1}},0,1): // plot with 0.1
```



```
>function f(x) := x^3-x; ...
>plot2d("f",r=1):
```



Berikut adalah ringkasan dari fungsi yang diterima

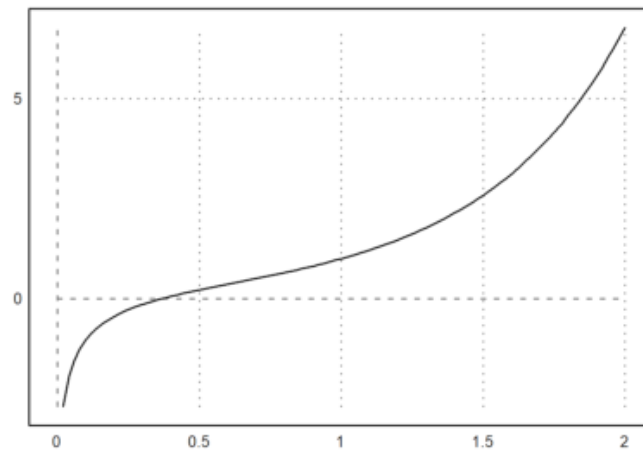
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolis. Untuk fungsi simbolis, nama saja yang berfungsi.

```
>function f(x) &= diff(x^x,x)
```

$$x^x (\log(x) + 1)$$

```
>plot2d(f,0,2):
```

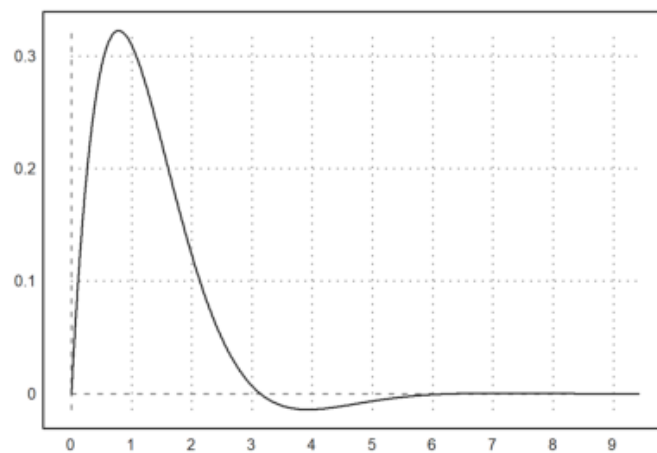


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.

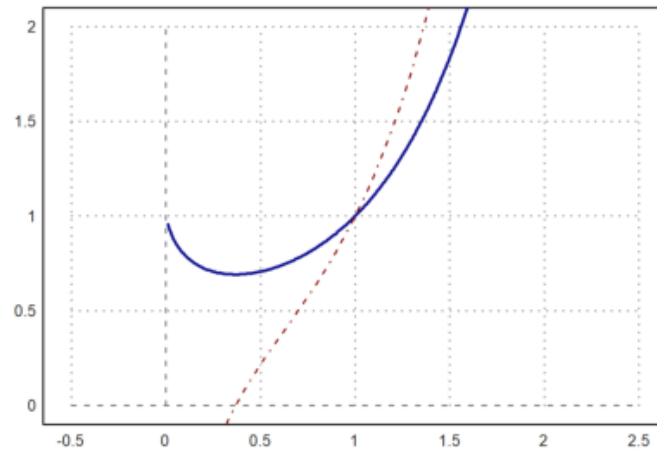
```
>expr &= sin(x)*exp(-x)
```

$$E^{-x} \sin(x)$$

```
>plot2d(expr,0,3pi):
```



```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```



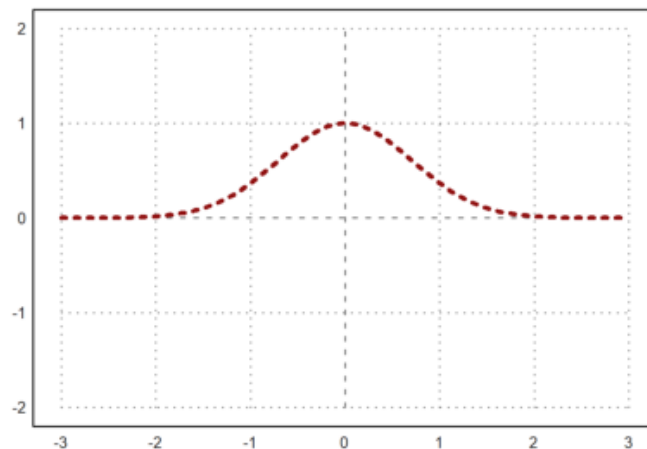
Untuk gaya garis ada berbagai pilihan.

- gaya="...". Pilih dari "-", "--", "-.", ".", ":", "-.-", "-.-".
- warna: Lihat di bawah untuk warna.
- ketebalan: Default adalah 1.

Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB.

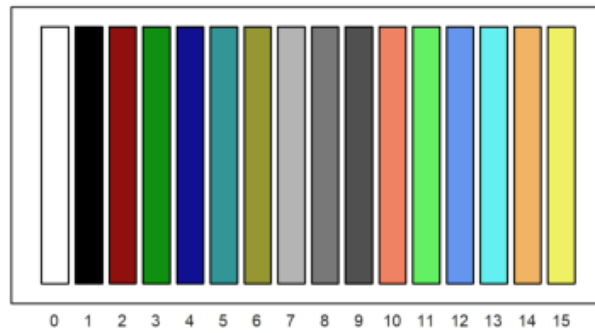
- 0.15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye terang, kuning
- rgb(merah, hijau, biru): parameter adalah real dalam [0,1].

```
>plot2d("exp(-x^2)",r=2,color=red,thickness=3,style="--") :
```



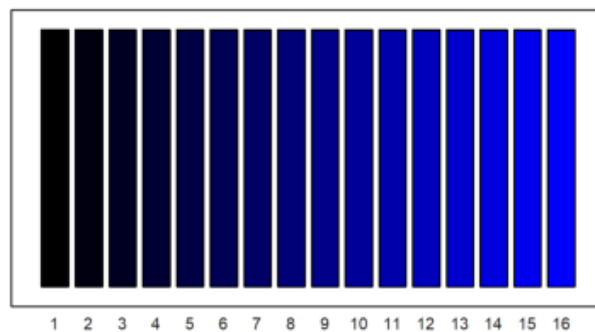
Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

```
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15) :
```



Tapi Anda bisa menggunakan warna apa saja.

```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```




---

**Catatan Baru**

---

**contoh penyelesaian masalah**

---

**1**

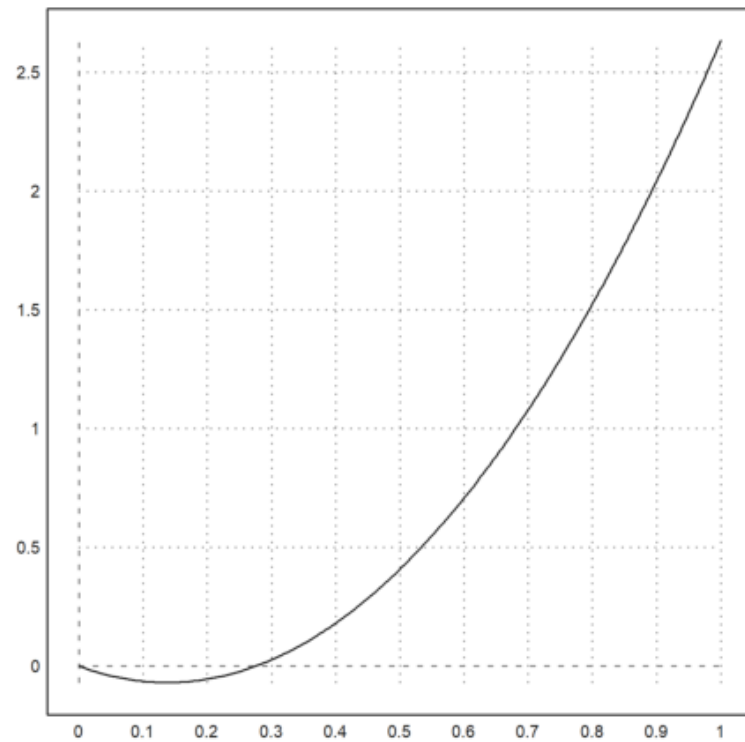
---

Menggambar Grafik Fungsi Kuadrat dengan Parameter

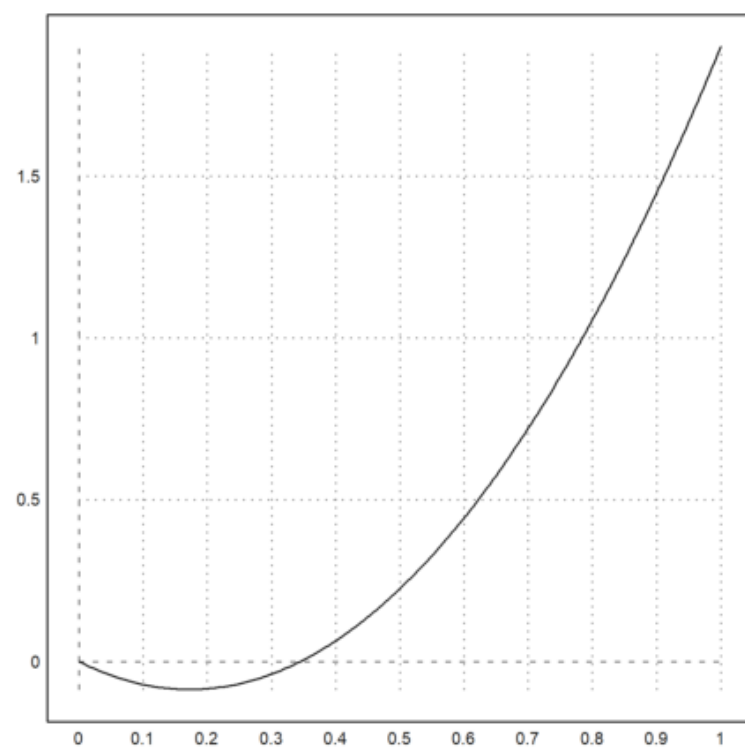
**Penyelesaian :**

---

```
>function f(x, a) := x^2/a + a*x^2 - x; // mendefinisikan fungsi
>a = 0.3; plot2d("f", 0, 1; a): // menggambar dengan a=0.3
```



```
>a = 0.4; plot2d("f", 0, 1; a): // menggambar dengan a=0.4
```





Penjelasan:

ungsi kuadrat diatur dengan parameter a. Dua grafik dihasilkan untuk dua nilai a, menunjukkan bagaimana parameter mempengaruhi bentuk kurva.

---

## contoh penyelesaian masalah menggambar kurva/plot 2D

---

2

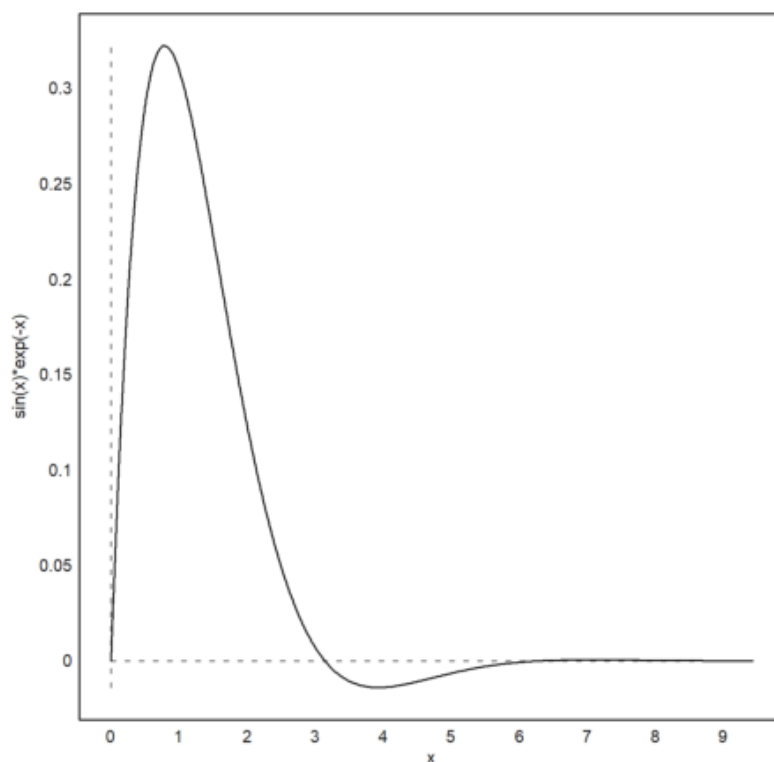
---

Menggambar Grafik Fungsi Trigonometri dengan Ekspresi Simbolik

**Penyelesaian :**

---

```
>expr &= sin(x) * exp(-x); // mendefinisikan ekspresi simbolik  
>plot2d(expr, 0, 3*pi, grid=1, xl="x", yl="sin(x)*exp(-x)"): // menggambar grafik
```



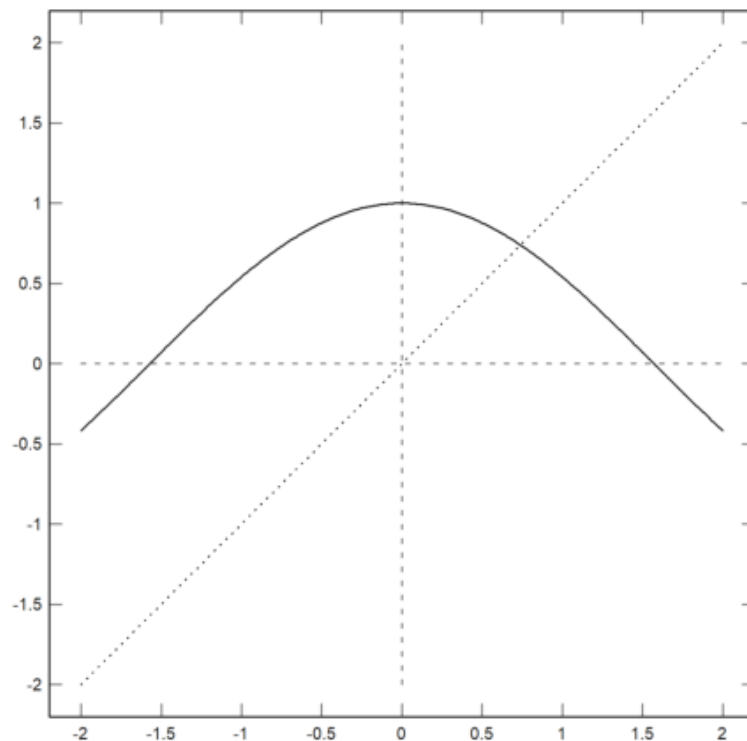
Penjelasan:

Grafik fungsi  $\sin(x) \cdot e^{-x}$  digambar dari 0 hingga  $3\pi$ , dengan grid untuk meningkatkan keterbacaan. Ekspresi simbolik memberikan fleksibilitas dalam mendefinisikan fungsi yang kompleks.

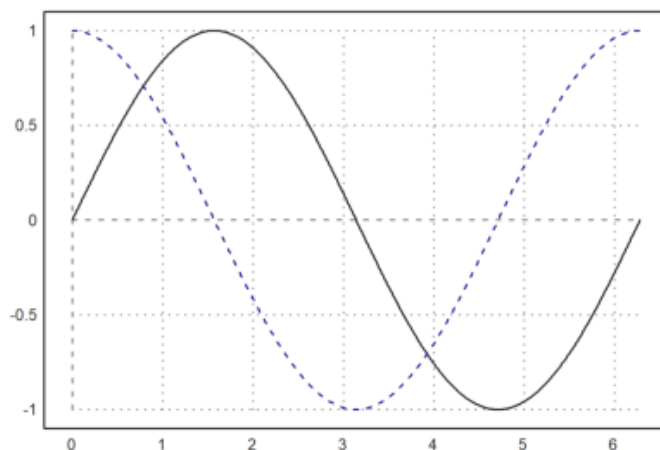
## Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metode menggunakan `>add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, tetapi panggilan pertama. Kami telah menggunakan fitur ini dalam contoh di atas.

```
>aspect(); plot2d("cos(x)",r=2,grid=6); plot2d("x",style=".",>add):
```

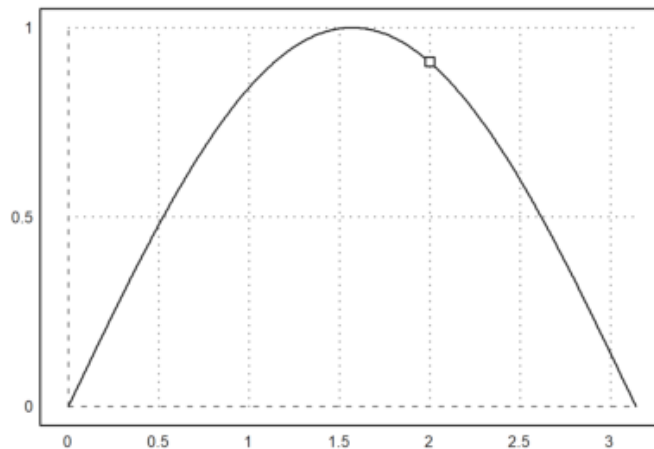


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



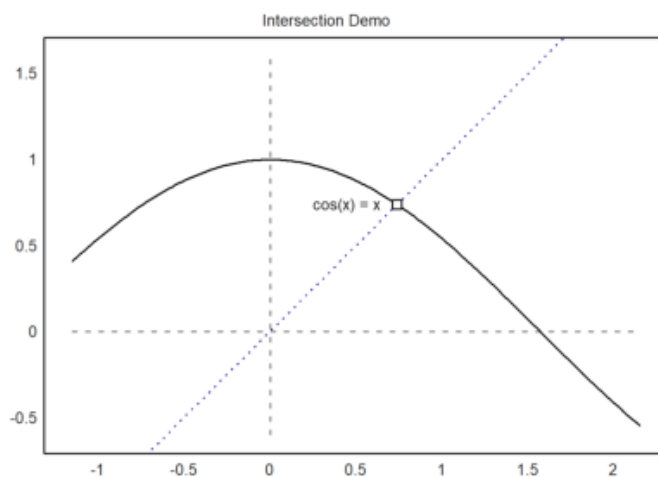
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



Kami menambahkan titik persimpangan dengan label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasilnya ke dalam notebook. Kami juga menambahkan judul ke plot.

```
>plot2d(["cos(x)", "x"], r=1.1, cx=0.5, cy=0.5, ...
> color=[black,blue], style=["-", "."], ...
> grid=1);
>x0=solve("cos(x)-x",1); ...
> plot2d(x0,x0,>points,>add,title="Intersection Demo"); ...
> label("cos(x) = x",x0,x0,pos="cl",offset=20):
```



Dalam demo berikut, kami memplot fungsi  $\text{sinc}(x) = \sin(x)/x$  dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolis.

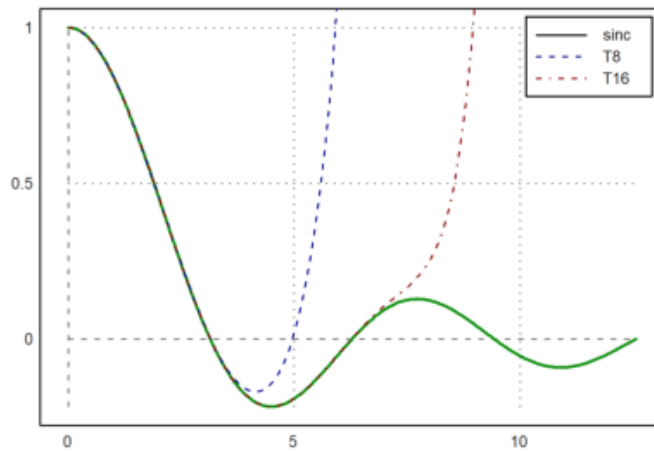
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Yang kedua dan yang ketiga memiliki set flag `>add`, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsi.

```
>$taylor(sin(x)/x,x,0,4)
```

$$\frac{x^4}{120} - \frac{x^2}{6} + 1$$

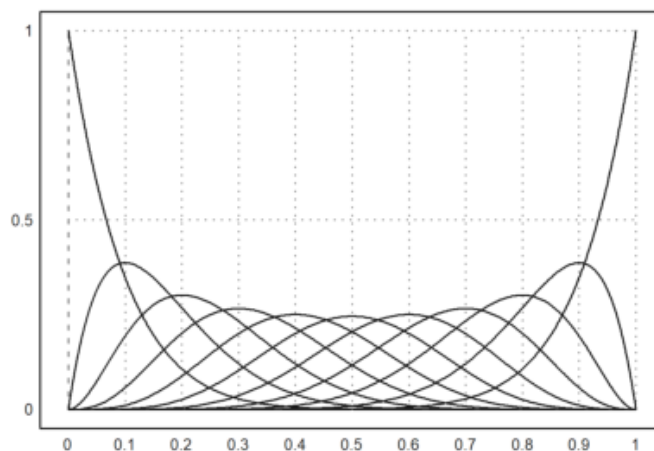
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
> plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
> plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
> labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
> colors=[black,blue,red]):
```



Dalam contoh berikut, kami menghasilkan Bernstein-Polinomial.

$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

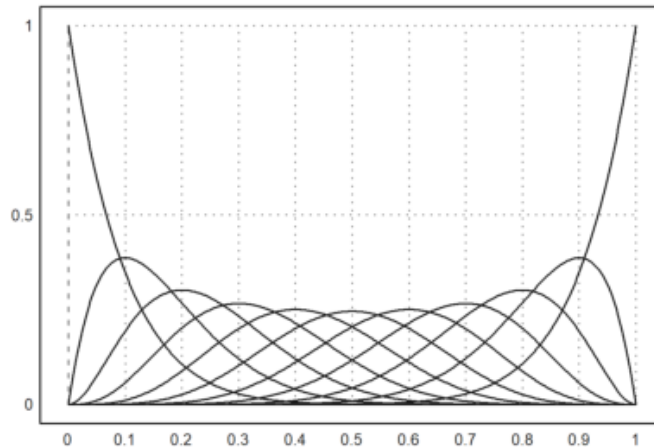
```
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



Metode kedua menggunakan pasangan matriks nilai-x dan matriks nilai-y yang berukuran sama.

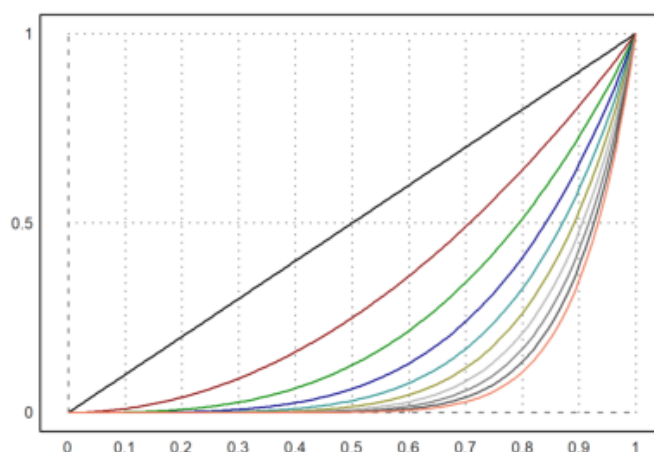
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

```
>x=linspace(0,1,500);  
>n=10; k=(0:n)'; // n is row vector, k is column vector  
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then  
>plot2d(x,y):
```



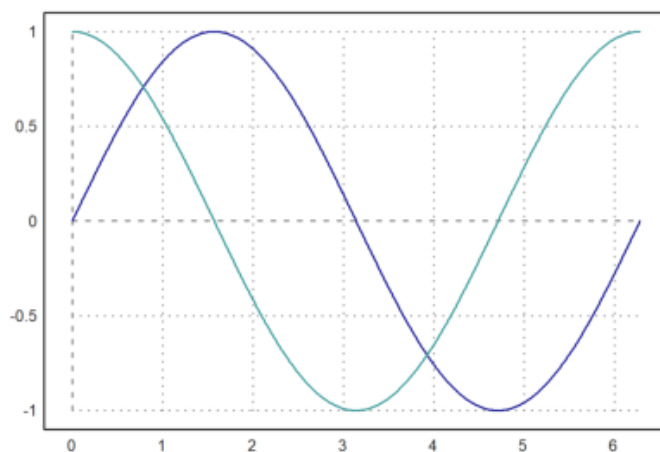
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
```

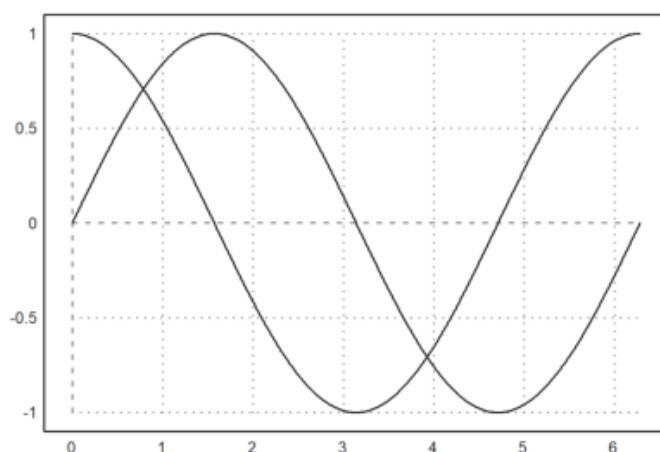


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

```
>plot2d(["sin(x)", "cos(x)"], 0, 2pi, color=4:5):
```



```
>plot2d(["sin(x)", "cos(x)"], 0, 2pi): // plot vector of expressions
```



Kita bisa mendapatkan vektor seperti itu dari Maxima menggunakan makelist() dan mxm2str().

```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

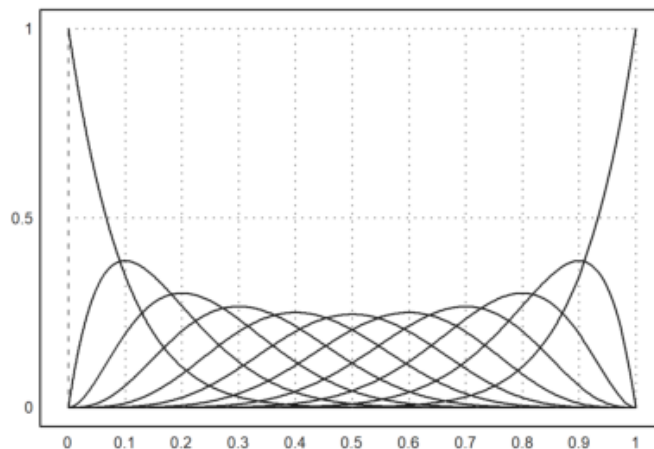
```

      10      9      8 2      7 3
[(1 - x) , 10 (1 - x) x, 45 (1 - x) x , 120 (1 - x) x ,
 6 4      5 5      4 6      3 7
210 (1 - x) x , 252 (1 - x) x , 210 (1 - x) x , 120 (1 - x) x ,
 2 8      9 10
45 (1 - x) x , 10 (1 - x) x , x ]
```

```
>mxm2str(v) // get a vector of strings from the symbolic vector
```

```
(1-x)^10  
10*(1-x)^9*x  
45*(1-x)^8*x^2  
120*(1-x)^7*x^3  
210*(1-x)^6*x^4  
252*(1-x)^5*x^5  
210*(1-x)^4*x^6  
120*(1-x)^3*x^7  
45*(1-x)^2*x^8  
10*(1-x)*x^9  
x^10
```

```
>plot2d(mxm2str(v),0,1): // plot functions
```

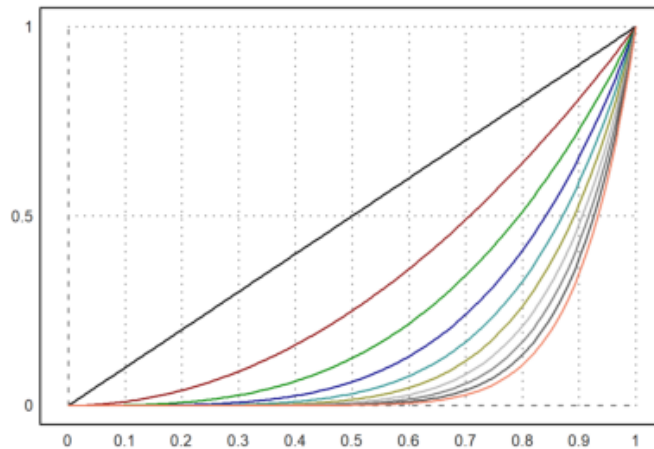


Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan diplot ke dalam satu plot.

Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika array warna ditambahkan, itu akan digunakan untuk setiap baris plot.

```
>n=(1:10)'; plot2d("x^n",0,1,color=1:10):
```

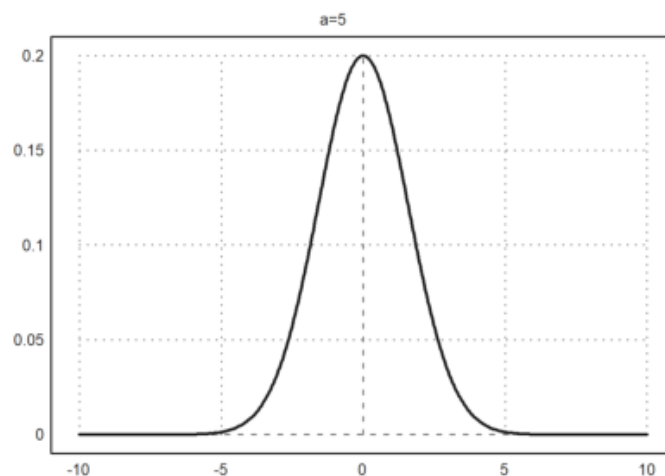


Ekspresi dan fungsi satu baris dapat melihat variabel global.

Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.

Berhati-hatilah, untuk meletakkan semua parameter yang ditetapkan di akhir perintah plot2d. Dalam contoh kita meneruskan  $a=5$  ke fungsi  $f$ , yang kita plot dari -10 hingga 10.

```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```

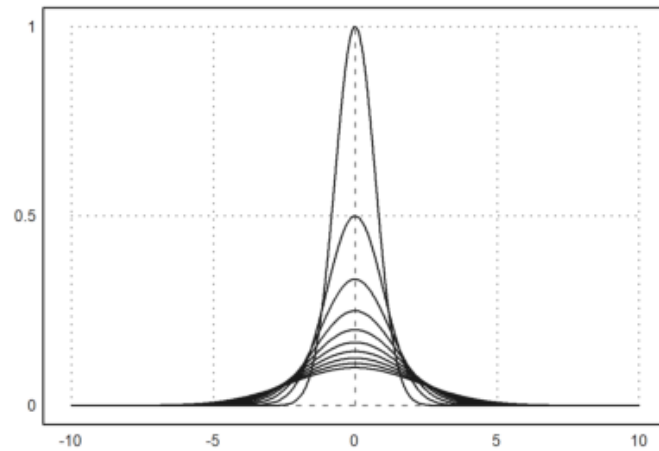


Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk meneruskan argumen ke fungsi yang dengan sendirinya diteruskan sebagai argumen ke fungsi lain.

Dalam contoh berikut, kami menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

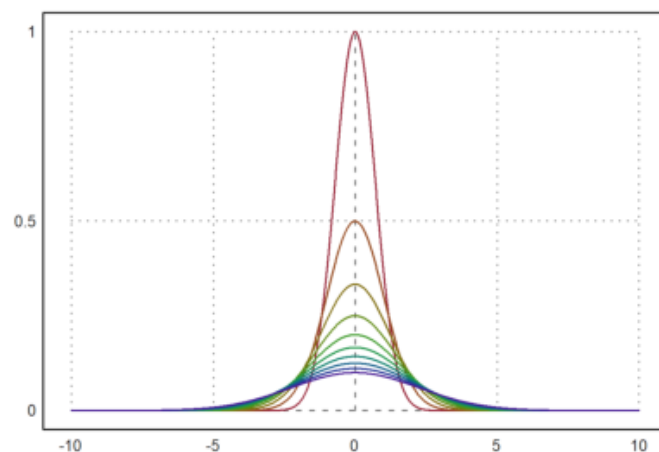
```
>plot2d({{"f",1}},-10,10); ...
>for a=2:10; plot2d({{"f",a}},>add); end:
```





Kami dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks  $f(x,a)$  adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi `getspectral()` untuk penjelasannya.

```
>x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10)):
```




---

**Catatan Baru**

---

**contoh penyelesaian masalah**

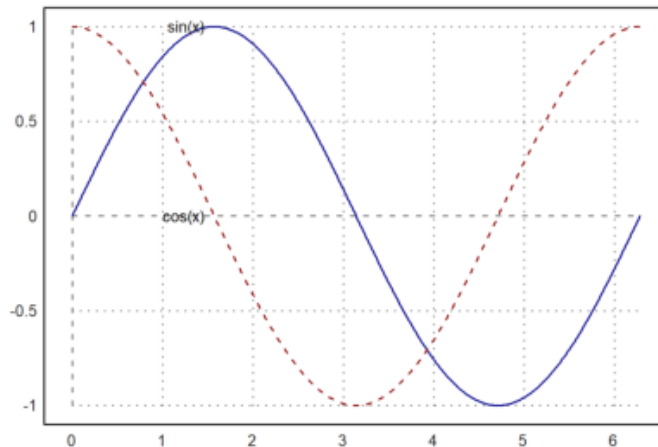
---

**1**

---

Menggambar Grafik Fungsi Trigonometri dengan Gaya Berbeda

```
>aspect(1.5); // Mengatur rasio aspek
>plot2d("sin(x)", 0, 2*pi, color=blue, style="-"); // Fungsi sin(x)
>plot2d("cos(x)", 0, 2*pi, color=red, style="--", >add); // Fungsi cos(x)
>label("sin(x)", pi/2, 1, pos="cl", offset=10); // Menambahkan label pada sin(x)
>label("cos(x)", pi/2, 0, pos="cl", offset=10): // Menambahkan label pada cos(x)
```



Penjelasan:

Grafik dari fungsi  $\sin(x)$  dan  $\cos(x)$  digambar dalam satu plot. Fungsi  $\sin(x)$  digambarkan dengan garis solid biru, sementara  $\cos(x)$  dengan garis putus-putus merah. Label ditambahkan untuk memperjelas identitas masing-masing fungsi.

---

2

---

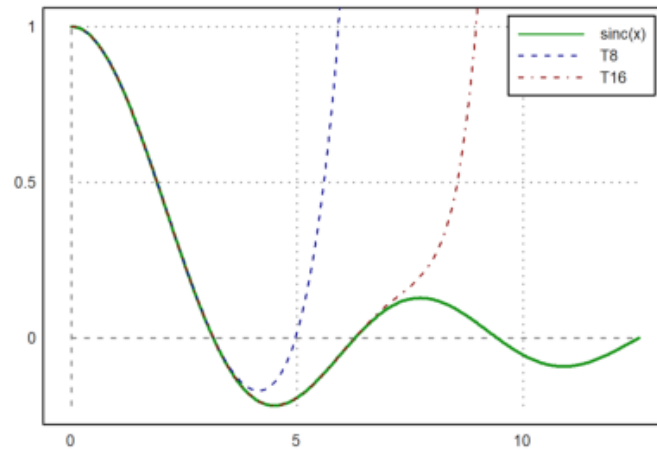
Menggambar Fungsi Sinc dan Ekspansi Taylor

---

Penyelesaian :

---

```
>plot2d("sinc(x)", 0, 4*pi, color=green, thickness=2); // Menggambar sinc(x)
>plot2d(&taylor(sin(x)/x, x, 0, 8), >add, color=blue, style="--"); // Ekspansi Taylor ke-8
>plot2d(&taylor(sin(x)/x, x, 0, 16), >add, color=red, style="-.-"); // Ekspansi Taylor ke-16
>labelbox(["sinc(x)", "T8", "T16"], styles=["-", "--", "-.-"], colors=[green, blue, red]):
```



Penjelasan:

Fungsi  $\text{sinc}(x)$  ditampilkan bersamaan dengan ekspansi Taylor derajat 8 dan 16. Setiap grafik diberi warna dan gaya yang berbeda, dan label kotak digunakan untuk menjelaskan fungsi yang digambarkan.

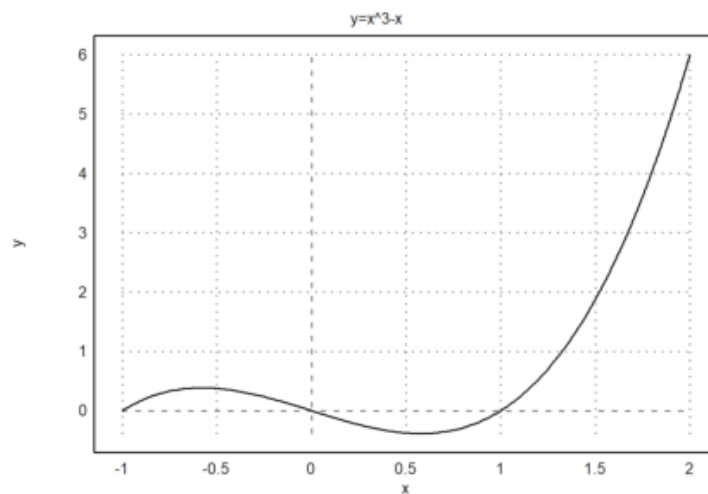
## Label Teks

Dekorasi sederhana bisa

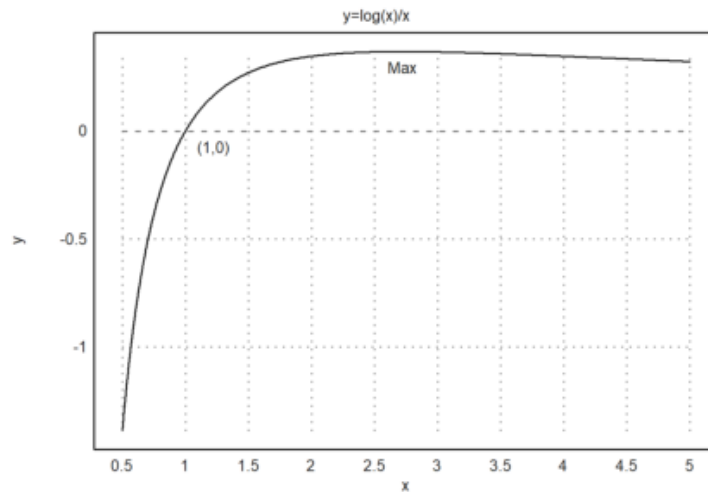
- judul dengan `judul="..."`
- x- dan y-label dengan `xl="...", yl="..."`
- label teks lain dengan `label("...",x,y)`

Perintah label akan memplot ke dalam plot saat ini pada koordinat plot (x,y). Itu bisa mengambil argumen posisi.

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x") :
```

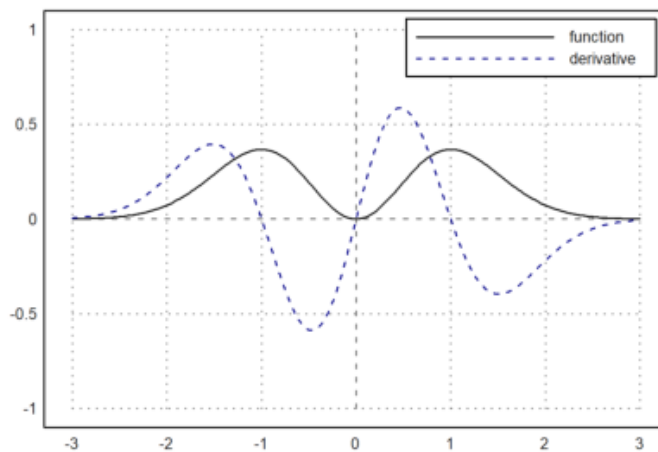


```
>expr := "log(x)/x"; ...
> plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



Ada juga fungsi `labelbox()`, yang dapat menampilkan fungsi dan teks. Dibutuhkan vektor string dan warna, satu item untuk setiap fungsi.

```
>function f(x) &= x^2*exp(-x^2); ...
>plot2d(&f(x),a=-3,b=3,c=-1,d=1); ...
>plot2d(&diff(f(x),x),>add,color=blue,style="--"); ...
>labelbox(["function","derivative"],styles=["-", "--"], ...
> colors=[black,blue],w=0.4):
```

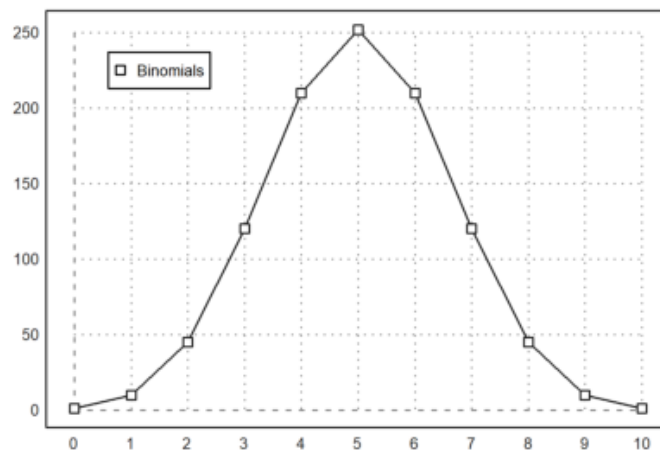


Kotak ditambahkan di kanan atas secara default, tetapi > kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebar nya otomatis.

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter >points, atau vektor flag, satu untuk setiap label.

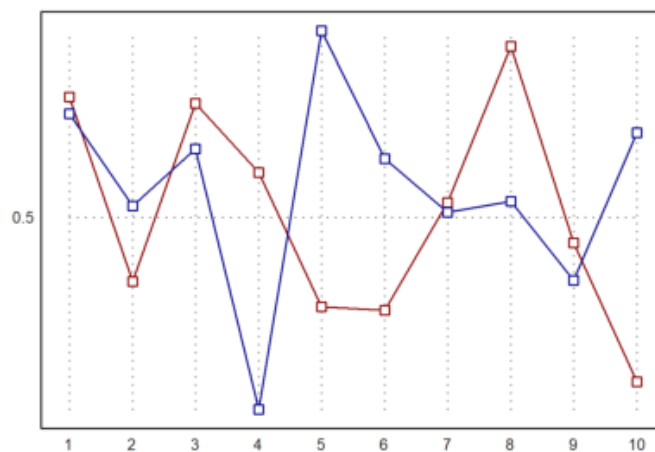
Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



Gaya plot ini juga tersedia di `statplot()`. Seperti di `plot2d()` warna dapat diatur untuk setiap baris plot. Ada lebih banyak plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

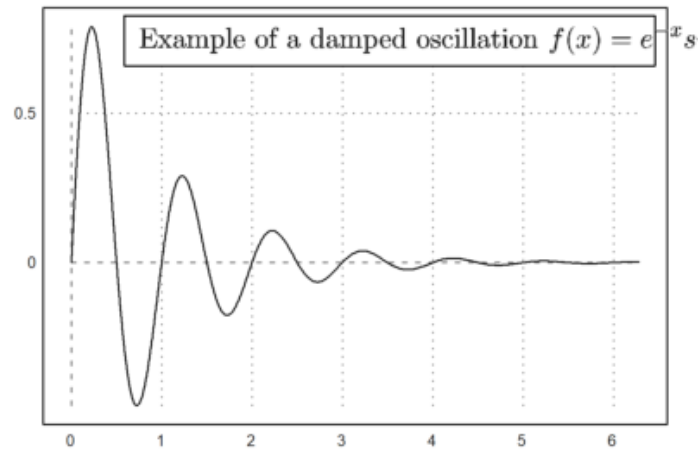
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi `textbox()`.

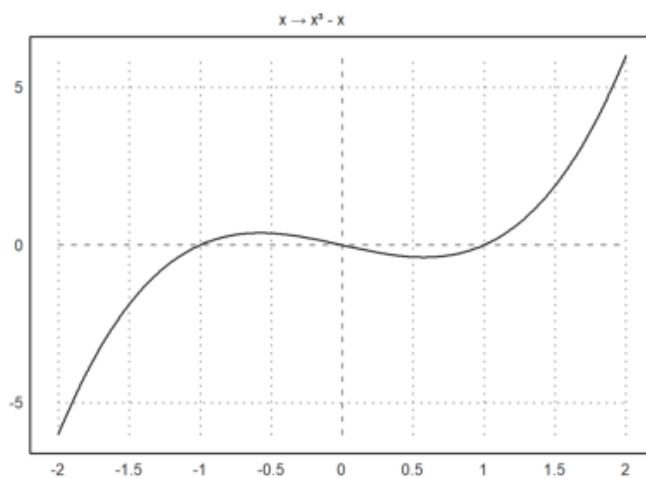
Lebar secara default adalah lebar maksimal dari baris teks. Tapi itu bisa diatur oleh pengguna juga.

```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
>plot2d("f(x)",0,2pi); ...
>textbox(latex("\text{Example of a damped oscillation}\ f(x)=e^{-x}\sin(2\pi x)"),w=0.85):
```



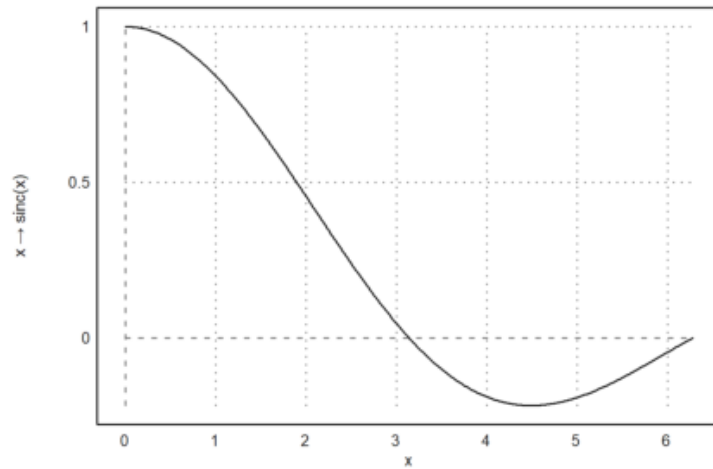
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

```
>plot2d("x^3-x",title=u"x → x³ - x"): 
```



Label pada sumbu x dan y bisa vertikal, begitu juga sumbunya.

```
>plot2d("sinc(x)",0,2pi,xl="x",yl=u"x → sinc(x)",>vertical): 
```




---

## Catatan Baru

---

MT memungkinkan kita untuk menambahkan elemen dekoratif pada grafik, seperti:

- Judul: Menggunakan parameter `title="..."`.
  - Label Sumbu: Menggunakan `xl="..."` untuk sumbu x dan `yl="..."` untuk sumbu y.
  - Label Teks: Menggunakan fungsi `label("...", x, y)` untuk menandai titik tertentu pada grafik.
  - Kotak Label: Fungsi `labelbox()` dapat digunakan untuk menampilkan teks dan informasi lainnya.
- 

## contoh penyelesaian masalah

---

1

---

Menggambar Fungsi Kuadrat dengan Label Sumbu dan Judul

---

Penyelesaian :

---

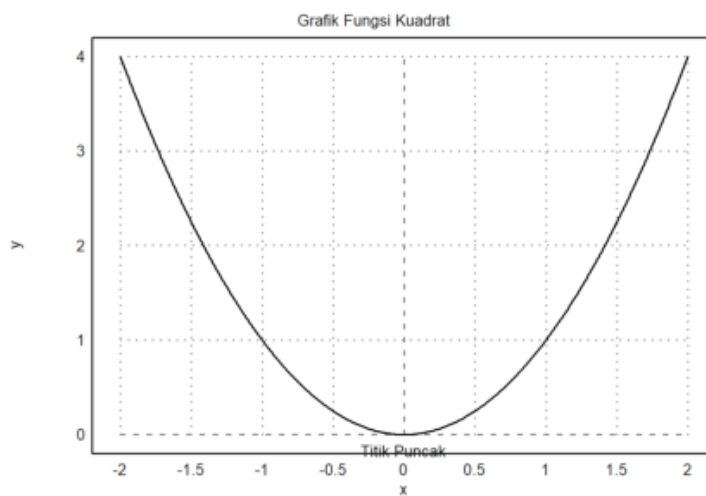
```
>plot2d("x^2", -2, 2, title="Grafik Fungsi Kuadrat", xl="x", yl="y");
```

```

Space between commands expected!
Found:
  (character 10)
You can disable this in the Options menu.
Error in:
... , title="Grafik Fungsi Kuadrat", xl="x", yl="y");
...

```

```
>label("Titik Puncak", 0, 0, pos="bc");
```



Penjelasan:

Grafik dari fungsi kuadrat  $y=x^2$  digambar dengan judul dan label untuk sumbu x dan y. Label juga ditambahkan untuk menandai titik puncak fungsi.

---

2

---

Menggambar Fungsi Logaritma dengan Label dan Kotak Label

---

**Penyelesaian :**

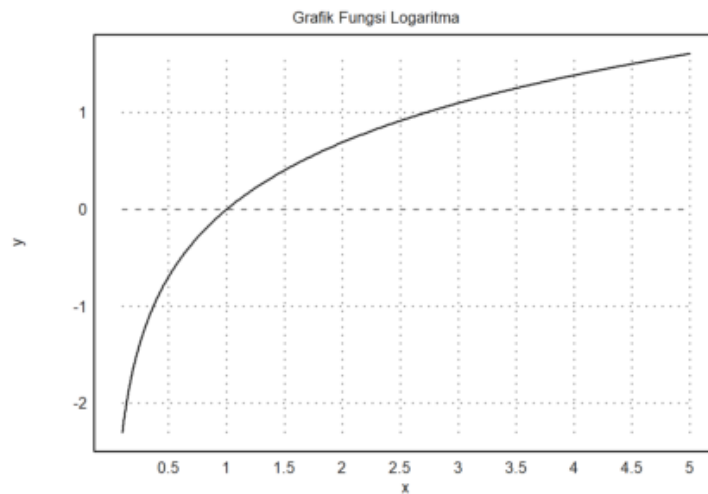
---

```

>expr := "log(x)";
>plot2d(expr, 0.1, 5, title="Grafik Fungsi Logaritma", xl="x", yl="y");
>label("Asimtot", 0, -10, pos="bc"); // Label asimtot di bawah sumbu
>labelbox("y = log(x)", styles="[]", x=0.7, y=2, tcolor=blue):

```





Penjelasan:

Grafik dari fungsi logaritma  $y=\log(x)$  digambar dengan judul dan label. Label untuk asimtot ditambahkan di bawah sumbu, serta kotak label yang menjelaskan fungsi yang digambarkan.

---

## LaTeX

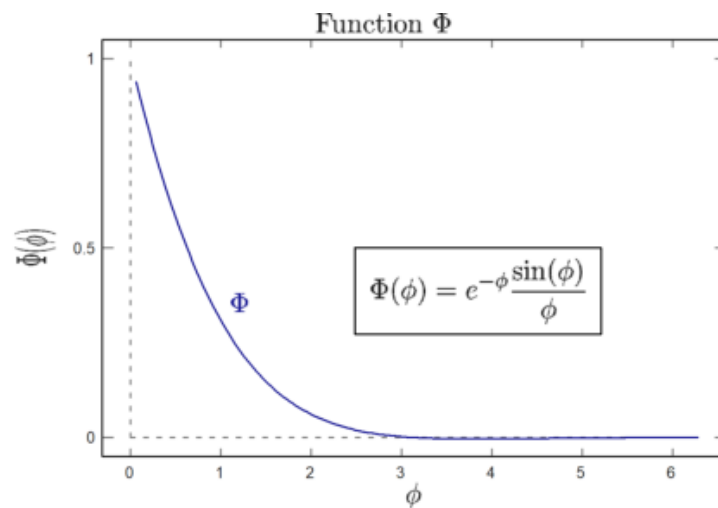
---

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "latex" dan "dvi2png" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan, bahwa penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil `latex()` sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

Dalam plot berikut, kami menggunakan LaTeX untuk label  $x$  dan  $y$ , label, kotak label, dan judul plot.

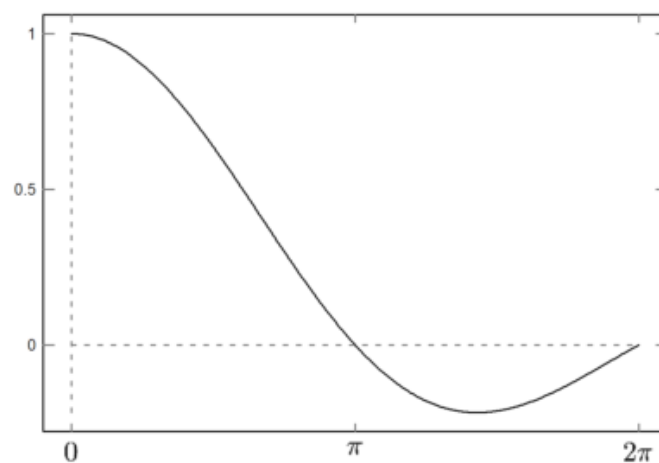
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
> title=latex("\text{Function } \Phi"), ...
> xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
> latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}",x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



Seringkali, kami menginginkan spasi dan label teks non-konformal pada sumbu x. Kita dapat menggunakan `xaxis()` dan `yaxis()` seperti yang akan kita tunjukkan nanti.

Cara termudah adalah dengan membuat plot kosong dengan bingkai menggunakan `grid=4`, lalu menambahkan grid dengan `ygrid()` dan `xgrid()`. Dalam contoh berikut, kami menggunakan tiga string LaTeX untuk label pada sumbu x dengan `xtick()`.

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
>ygrid(-2:0.5:2,grid=6); ...
>xgrid([0:2]*pi,<ticks,grid=6); ...
>xtick([0,pi,2pi],["0","\pi","2\pi"],>latex):
```



Tentu saja, fungsi juga dapat digunakan.

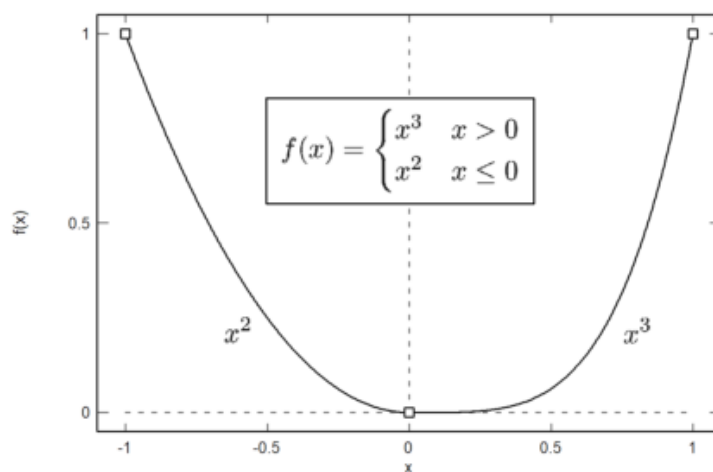
```
>function map f(x) ...

  if x>0 then return x^4
  else return x^2
  endif
endfunction
```

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tetapi untuk mendemonstrasikan vektorisasi itu berguna, kami menambahkan beberapa poin kunci ke plot di  $x=-1$ ,  $x=0$  dan  $x=1$ .

Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakannya untuk dua label dan kotak teks. Tentu saja, Anda hanya akan dapat menggunakan LaTeX jika Anda telah menginstal LaTeX dengan benar.

```
>plot2d("f",-1,1,xl="x",yl="f(x)",grid=6); ...
>plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
>label(latex("x^3"),0.72,f(0.72)); ...
>label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
>textbox( ...
> latex("f(x)=\begin{cases} x^3 & x>0 \\ x^2 & x \le 0 \end{cases}"), ...
> x=0.7,y=0.2):
```




---

**Catatan Baru**

---

**contoh penyelesaian masalah**

---

**1**

---

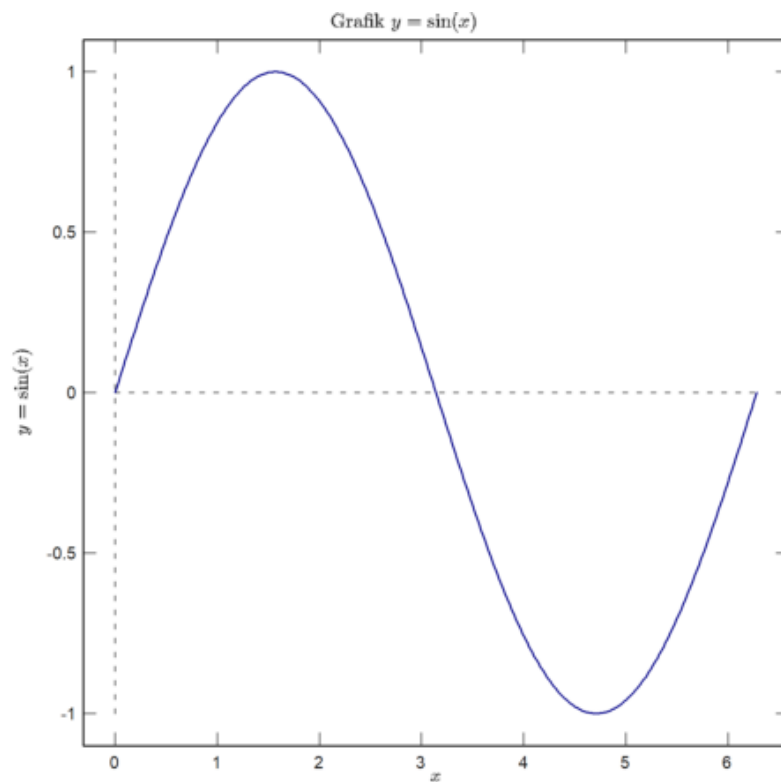
Buat grafik fungsi  $y=\sin(x)$  pada interval  $[0,2\pi]$ . Tambahkan judul dan label sumbu.

---

**Penyelesaian :**

---

```
>plot2d("sin(x)", 0, 2*pi, grid=6, color=blue, title=latex("\text{Grafik } y = \sin(x)"), x1=0, x2=2*pi, y1=-1, y2=1)
```




---

2

---

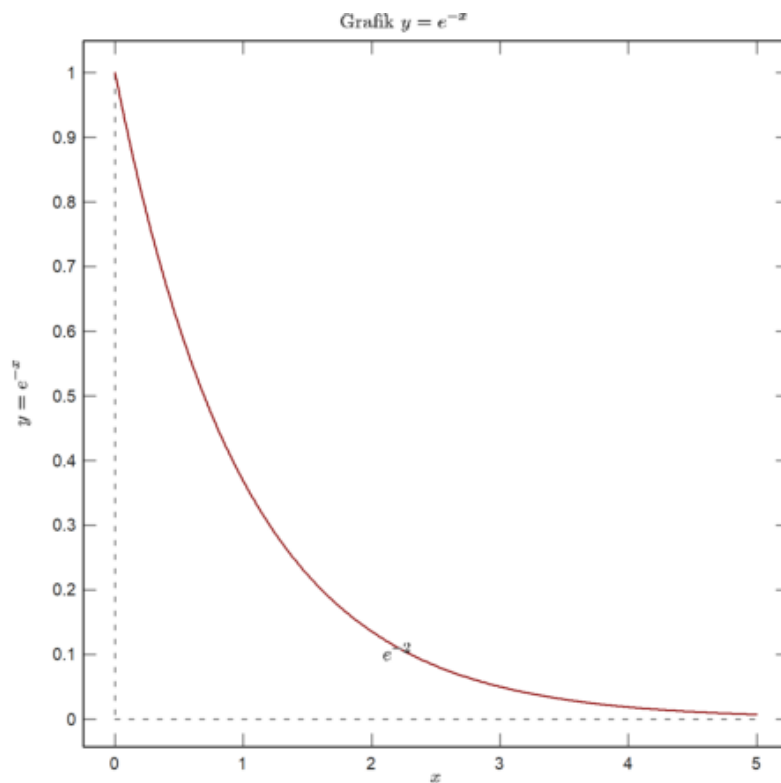
Gambarlah grafik fungsi  $y=e^{-x}$  pada interval  $[0,5]$ . Sertakan label yang menunjukkan nilai fungsi pada  $x=2$ .

---

**Penyelesaian :**

---

```
>plot2d("exp(-x)", 0, 5, grid=6, color=red, title=latex("\text{Grafik } y = e^{-x}"), x1=0, x2=5, y1=0, y2=1.1)
>label(latex("e^{-2}"), 2, exp(-2)):
```




---

## Interaksi pengguna

---

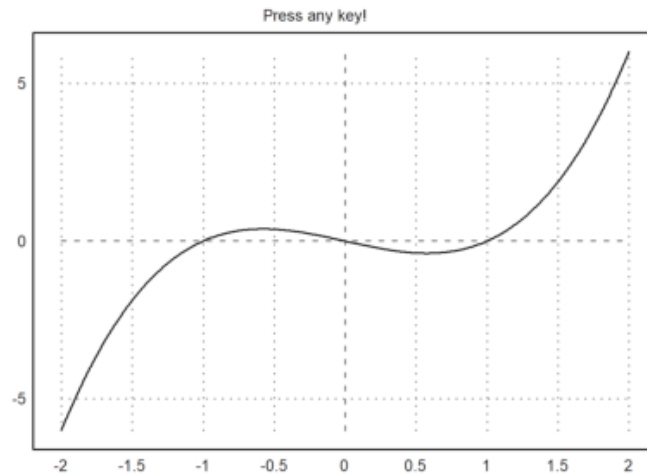
Saat memplot fungsi atau ekspresi, parameter `>user` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

- perbesar dengan `+` atau `-`
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali

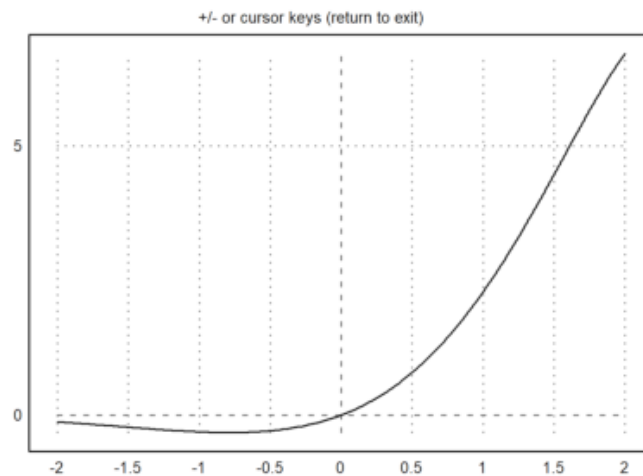
Tombol spasi akan mengatur ulang plot ke jendela plot asli.

Saat memplot data, flag `>user` hanya akan menunggu penekanan tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!"):
```



```
>plot2d("exp(x)*sin(x)",user=true, ...
> title="+/- or cursor keys (return to exit)");
```



Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan `mousedrag()` menunggu event mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse dipindahkan atau mouse ke atas, dan penekanan tombol. Fungsi `dragpoints()` memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.

Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

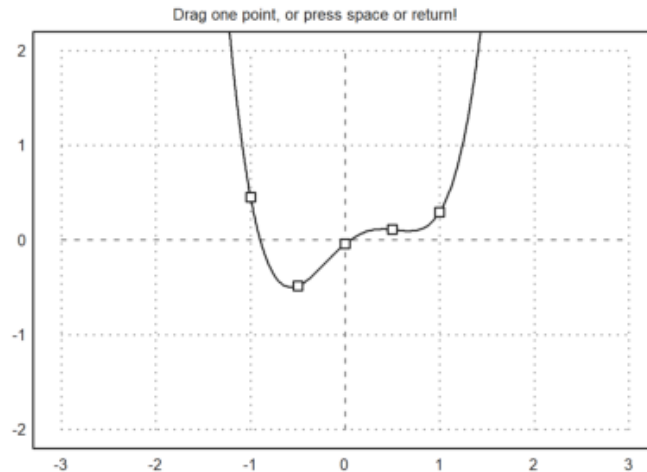
```
>function plotf(xp,yp,select) ...

    d=interp(xp,yp);
    plot2d("interpval(xp,d,x)";d,xp,r=2);
    plot2d(xp,yp,>points,>add);
    if select>0 then
        plot2d(xp[select],yp[select],color=red,>points,>add);
    endif;
    title("Drag one point, or press space or return!");
endfunction
```

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
```



Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

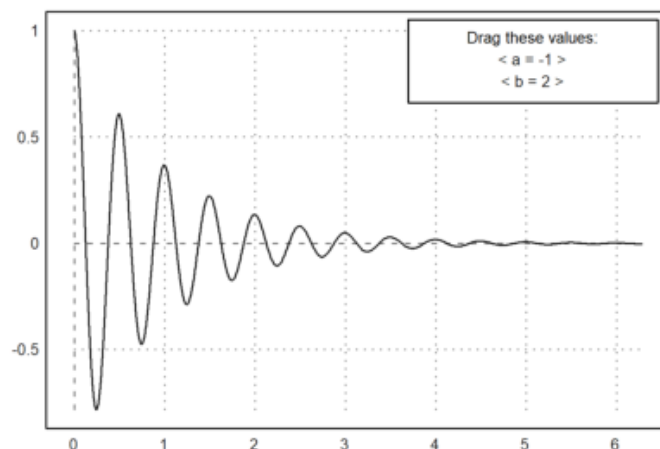
Pertama kita membutuhkan fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
```

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.

Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
>dragvalues("plotf",["a","b],[-1,2],[[-2,2];[1,10]], ...  
> heading="Drag these values:",hcolor=black):
```



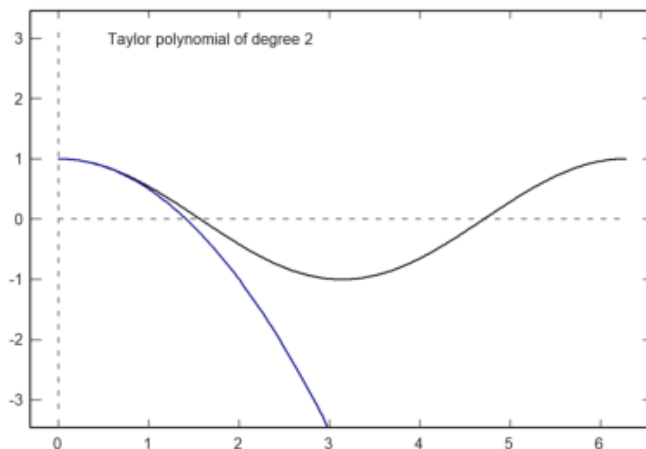
Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>function plotf(n) ...
```

```
    plot2d("cos(x)",0,2pi,>square,grid=6);  
    plot2d("&taylor(cos(x),x,0,@n)",color=blue,>add);  
    textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);  
endfunction
```

Sekarang kami mengizinkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam buku catatan.

```
>nd=dragvalues("plotf","degree",2,[0,20],20,y=0.8, ...  
> heading="Drag the value:"); ...  
>plotf(nd):
```



Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak poin.

```
>function dragtest ...
```

```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");  
    start=0;  
    repeat  
        {flag,m,time}=mousedrag();  
        if flag==0 then return; endif;  
        if flag==2 then  
            hold on; mark(m[1],m[2]); hold off;  
        endif;  
    end  
endfunction
```



```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

---

**Catatan Baru**

---

**contoh penyelesaian masalah**

---

**1**

---

Buatlah grafik fungsi  $y=x^3-ax$  pada interval  $[-1,1]$  dengan parameter  $a$  yang dapat disesuaikan. Biarkan pengguna dapat menyeret titik di grafik.

---

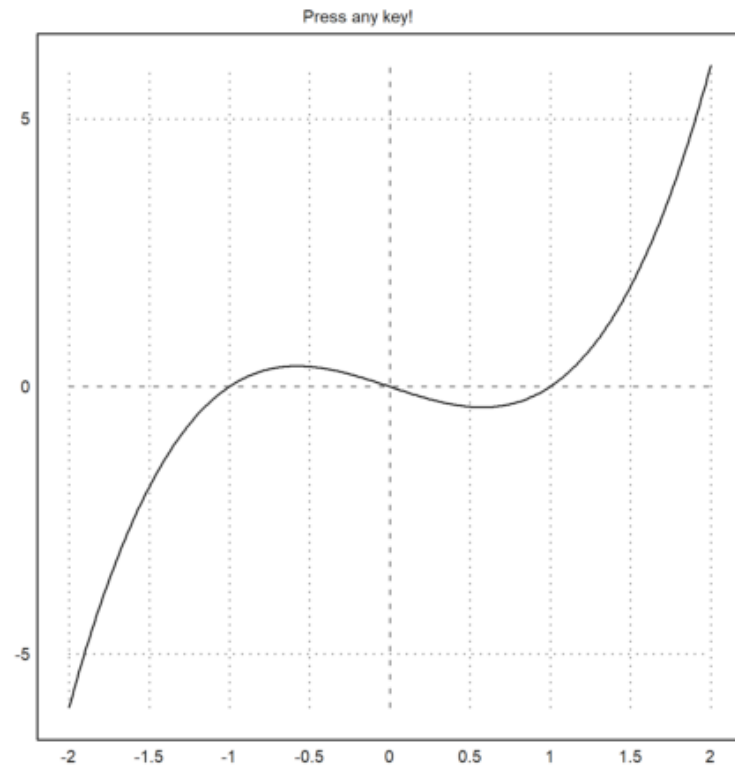
**Penyelesaian :**

---

```
>function plotf(a) ...
```

```
    plot2d("x^3 - a*x", -1, 1, grid=6, title=latex("\text{Grafik } y = x^3 - ax"));
endfunction
```

```
>plot2d({"x^3-a*x",a=1}), >user, title="Press any key!":
```



```
>dragpoints("plotf", [1]):
```

Function dragpoints needs at least 3 arguments!  
 Use: dragpoints (f\$: string, x: real, y: real {, status})  
 Error in:  
 dragpoints("plotf", [1]): ...  
                                   ^

---

2

---

Gambarlah grafik fungsi  $y=\sin(ax)$  pada interval  $[0,2\pi]$ , dengan slider untuk parameter  $a$  yang memungkinkan pengguna untuk mengubah frekuensi.

---

**Penyelesaian :**

---

```
>function plotf(a) ...
```

```
plot2d("sin(a*x)", 0, 2*pi, grid=6, title=latex("\text{Grafik } y = \sin(a x)"));
endfunction
```

```
>dragvalues("plotf", ["a"], [1], [[0.5, 5]], heading="Adjust the frequency:", hcolor=black
```

```
Variable a not found!
Use global variables or parameters for string evaluation.
Error in expression: sin(a*x)
%ploteval:
  y0=f$(x[1],args());
adaptiveevalone:
  s=%ploteval(g$,t;args());
plot2d:
  dw/n,dw/n^2,dw/n,auto;args());
plotf:
  plot2d("sin(a*x)", 0, 2*pi, grid=6, title=latex("\text{Grafik ...
Try "trace errors" to inspect local variables after errors.
dragvalues:
  f$(vv,args());
```

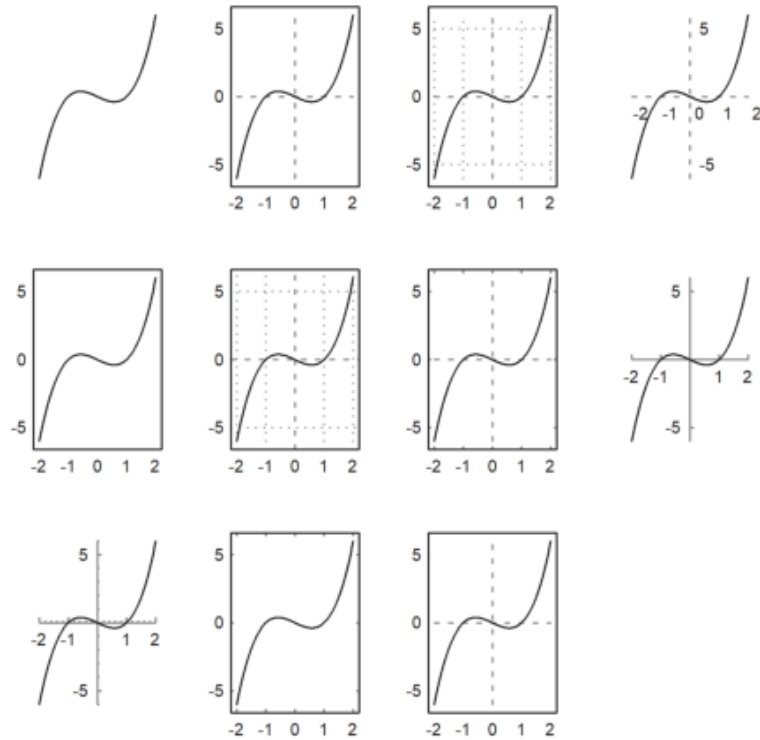
---

## Gaya Plot 2D

---

Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

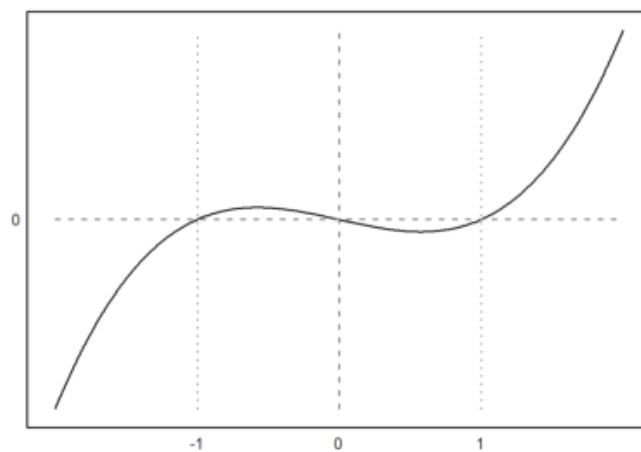
```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("x^3-x",grid=4); ... // no ticks, only labels
> figure(6); plot2d("x^3-x",grid=5); ... // default, but no margin
> figure(7); plot2d("x^3-x",grid=6); ... // axes only
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis
> figure(9); plot2d("x^3-x",grid=8); ... // axes only, finer ticks at axis
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks inside
> figure(11); plot2d("x^3-x",grid=10); ...// no ticks, axes only
> figure(0):
```



Parameter `<frame` mematikan frame, dan `framecolor=blue` mengatur frame ke warna biru.

Jika Anda ingin centang sendiri, Anda dapat menggunakan `style=0`, dan menambahkan semuanya nanti.

```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid
```

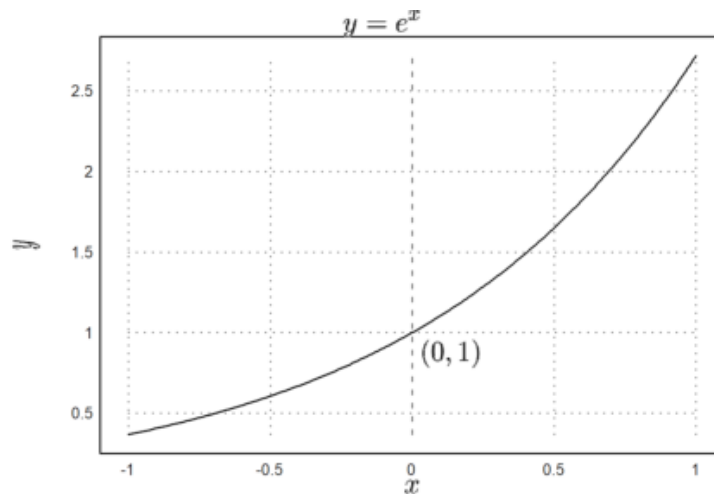


Untuk judul plot dan label sumbu, lihat contoh berikut.

```

>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)"),0,1,color=blue): // label a point

```

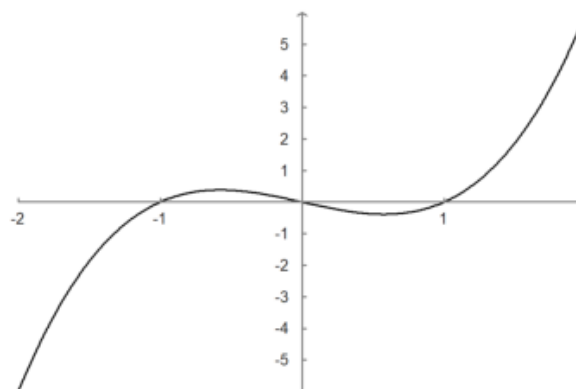


Sumbu dapat digambar secara terpisah dengan `xaxis()` dan `yaxis()`.

```

>plot2d("x^3-x",<grid,<frame);
>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->"):

```



Teks pada plot dapat diatur dengan `label()`. Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

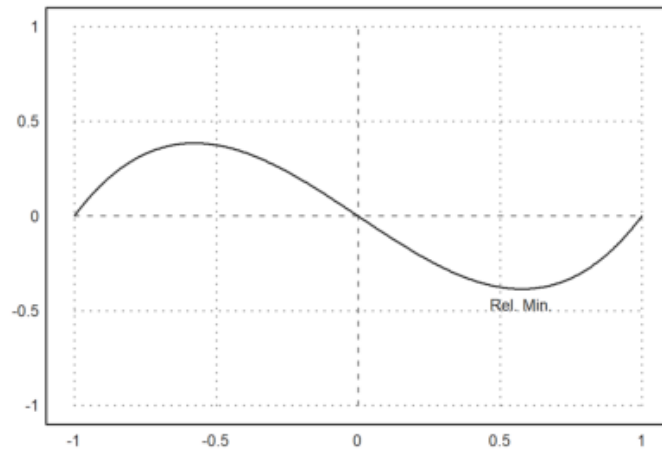
```

>function f(x) &= x^3-x

```

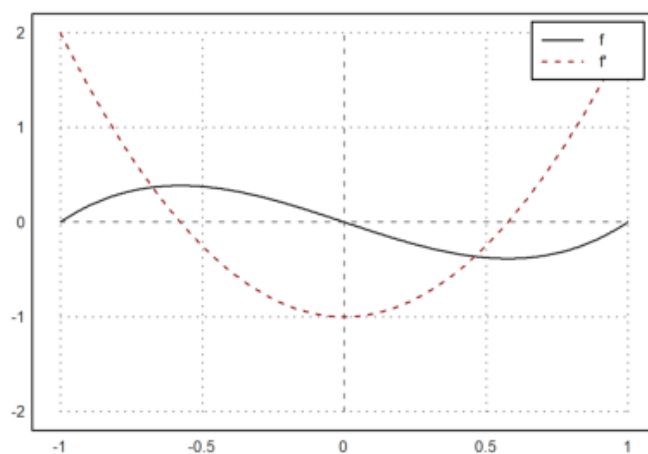
$$3 \\ x - x$$

```
>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // compute point of minimum
>label("Rel. Min.",x0,f(x0),pos="lc"): // add a label there
```

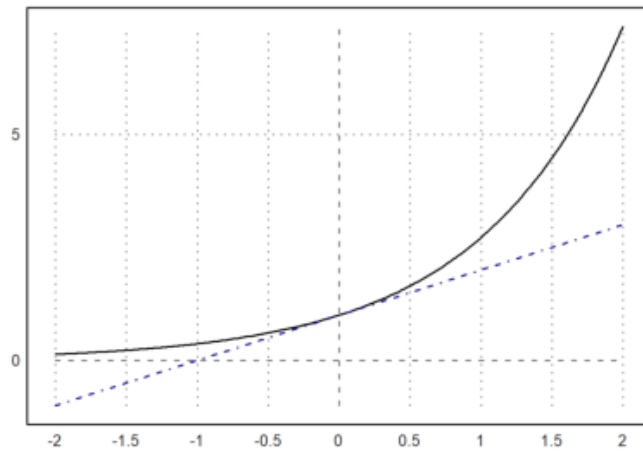


Ada juga kotak teks.

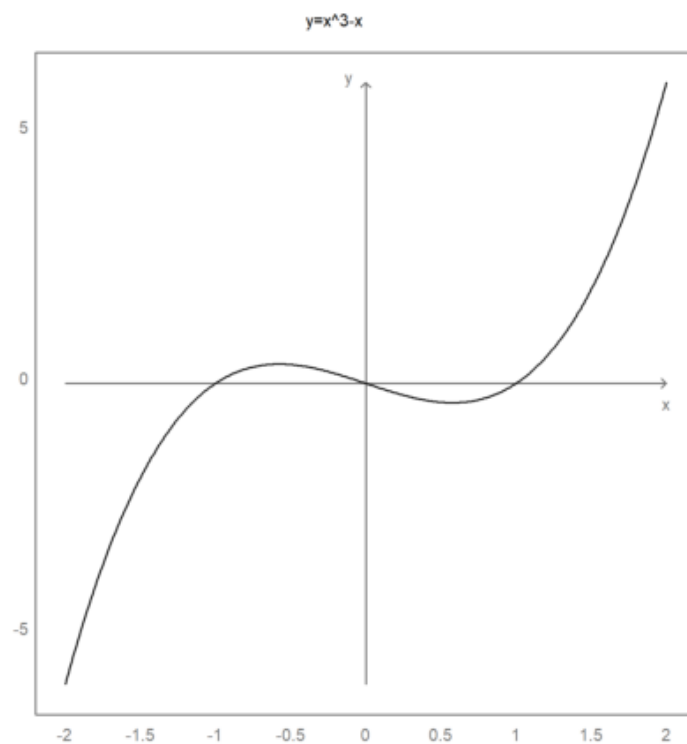
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-", "--"],[black,red]): // label box
```



```
>plot2d(["exp(x)", "1+x"],color=[black,blue],style=["-", "-.-"]):
```



```
> gridstyle("-", color=gray, textcolor=gray, framecolor=gray); ...
> plot2d("x^3-x", grid=1); ...
> settitle("y=x^3-x", color=black); ...
> label("x", 2, 0, pos="bc", color=gray); ...
> label("y", 0, 6, pos="cl", color=gray); ...
> reset():
```



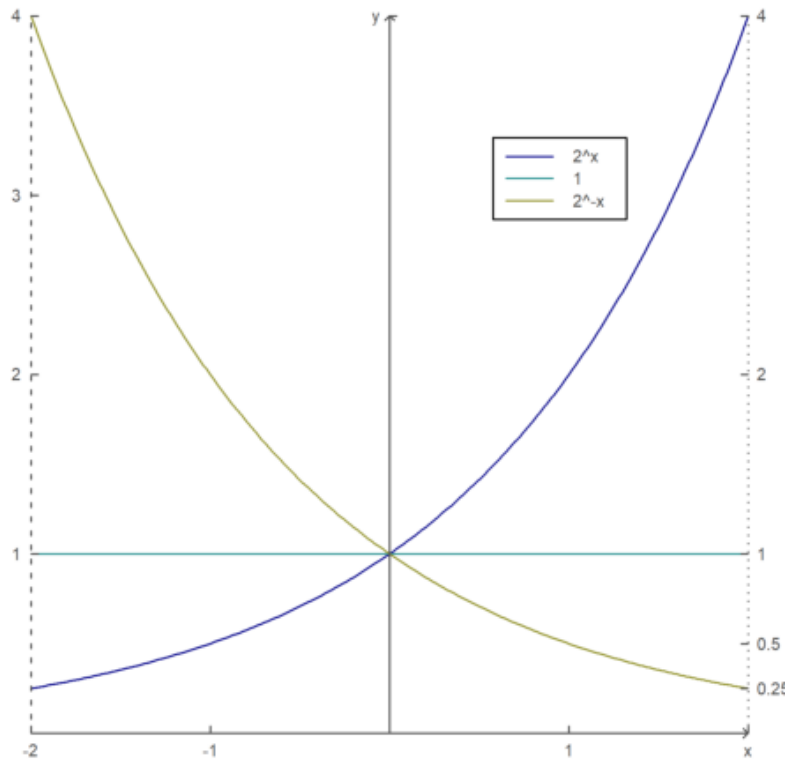
Untuk kontrol lebih, sumbu x dan sumbu y dapat dilakukan secara manual.

Perintah `fullwindow()` memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan `shrinkwindow()` atau `reset()` untuk mengatur ulang ke default.

```

>fullwindow; ...
> gridstyle(color=darkgray,textcolor=darkgray); ...
> plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
> xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
> yaxis(0,4,"y",style="->"); ...
> yaxis(-2,1:4,>left); ...
> yaxis(2,2^(-2:2),style=".",<left); ...
> labelbox(["2^x","1","2^-x"],colors=4:6,x=0.8,y=0.2); ...
> reset:

```



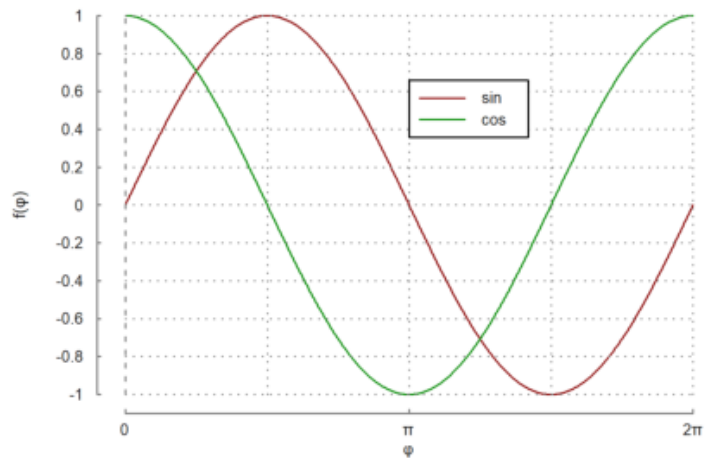
Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbu di luar area plot.

```

>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0","u"&pi;","u"2&pi;"],style="-",>ticks,>zero); ...
> xgrid((0:0.5:2)*pi,<ticks); ...
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left); ...
> xlabel(u"&pi;"); ylabel(u"f(&pi;)"):

```






---

## Catatan Baru

---

### contoh penyelesaian masalah

---

1

---

Gambarlah grafik fungsi  $y=x^3-x$  dengan sumbu yang disesuaikan dan label. Buatlah judul untuk plot tersebut.

---

## Penyelesaian :

---

```
>function f(x) := x^3 - x
>plot2d(f, -2, 2, grid=1, frame=true);
```

```
Function f needs at least one argument!
Use: f (x)
Error in:
plot2d(f, -2, 2, grid=1, frame=true); ...
      ^
```

```
>title(latex("Grafik Fungsi $y = x^3 - x$"));
```

```
Error with Latex formula
Grafik Fungsi $y = x^3 - x$
texpng:
  return _texpng(latex,usealias,factor,backcolor,color);
Try "trace errors" to inspect local variables after errors.
latex:
```

```

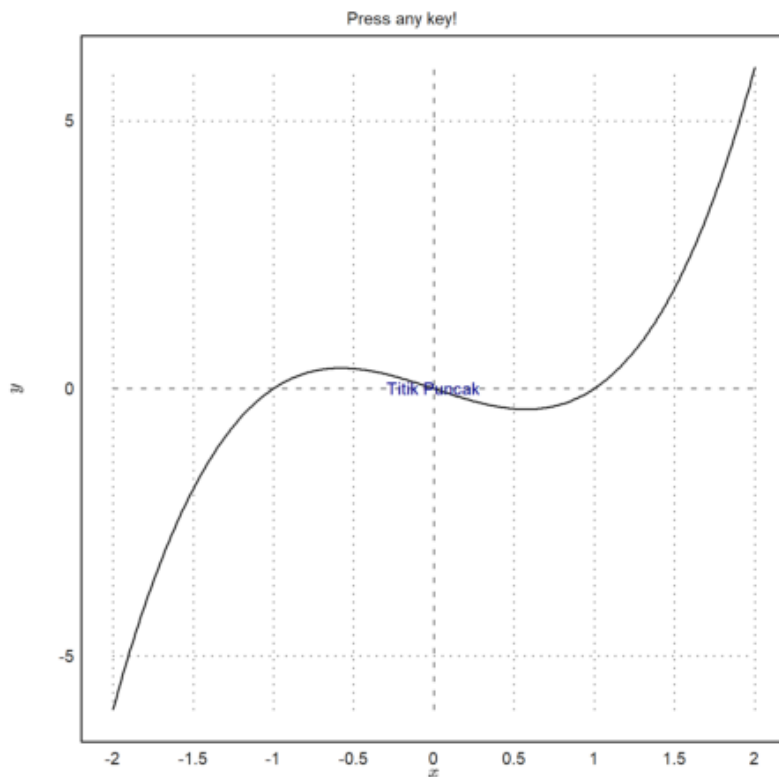
return loadrgb(texpng(s,=color,=factor));
Error in:
title(latex("Grafik Fungsi  $y = x^3 - x$ ")); ...
^

```

```

>xlabel(latex("x"));
>ylabel(latex("y"));
>label("Titik Puncak", 0, 0, pos="cc", color=blue):

```




---

2

---

Gambarlah grafik fungsi  $y=e^x$  pada interval  $[-1,1]$ , dan tambahkan label pada titik  $(0, 1)$ . Sertakan juga judul dan label sumbu.

---

**Penyelesaian :**

---

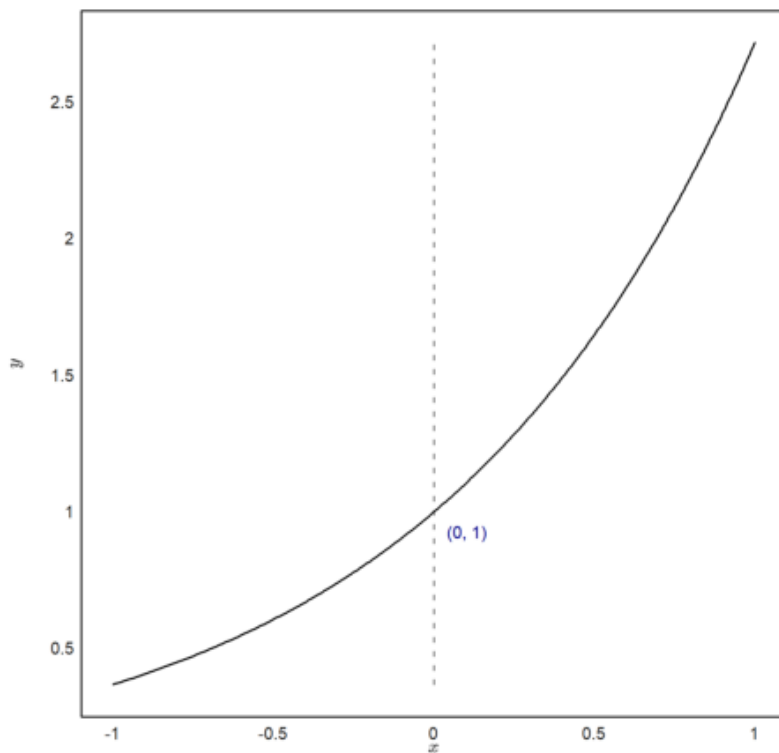
```

>plot2d("exp(x)", -1, 1, grid=1, frame=true);
>title(latex("Grafik Fungsi  $y = e^x$ "));

```

```
Error with Latex formula
Grafik Fungsi  $y = e^x$ 
texpng:
    return _texpng(latex, usealias, factor, bgcolor, color);
Try "trace errors" to inspect local variables after errors.
latex:
    return loadrgb(texpng(s, =color, =factor));
Error in:
title(latex("Grafik Fungsi  $y = e^x$ ")); ...
^
```

```
>xlabel(latex("x"));
>ylabel(latex("y"));
>label("(0, 1)", 0, 1, color=blue):
```




---

## Merencanakan Data 2D

---

Jika  $x$  dan  $y$  adalah vektor data, data ini akan digunakan sebagai koordinat  $x$  dan  $y$  dari suatu kurva. Dalam hal ini,  $a$ ,  $b$ ,  $c$ , dan  $d$ , atau radius  $r$  dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, >persegi dapat diatur untuk menjaga rasio aspek persegi.

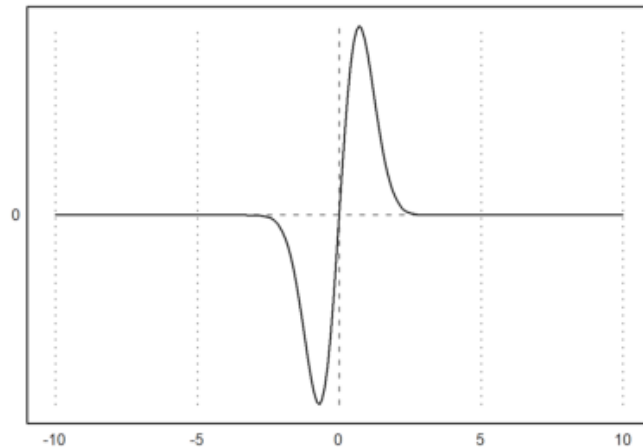
Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai  $x$ , dan satu atau beberapa baris nilai  $y$ . Dari rentang dan nilai- $x$ , fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan ">titik", untuk garis campuran dan titik gunakan ">tambahan".

Tapi Anda bisa memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y):
```



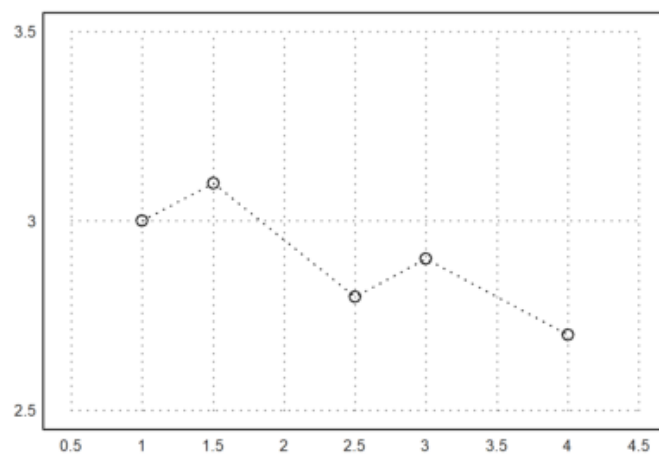
Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

- style="...": Pilih dari "[", "<>", "o", ":", ":", "+", "\*", "[]", "<>", "o", ":", ":", ":", "|".

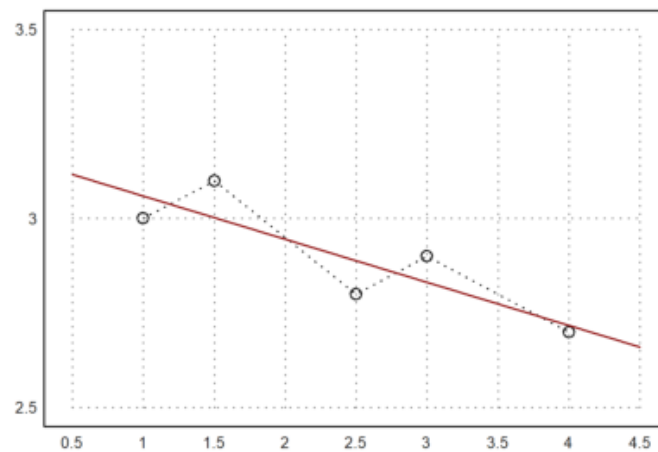
Untuk memplot set poin gunakan >points. Jika warna adalah vektor warna, setiap titik mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.

Parameter >addpoints menambahkan titik ke segmen garis untuk plot data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // lines
>plot2d(xdata,ydata,>points,>add,style="o"): // add points
```



```
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red): // add plot of line
```




---

## Catatan Baru \*\* contoh penyelesaian masalah

---

1

---

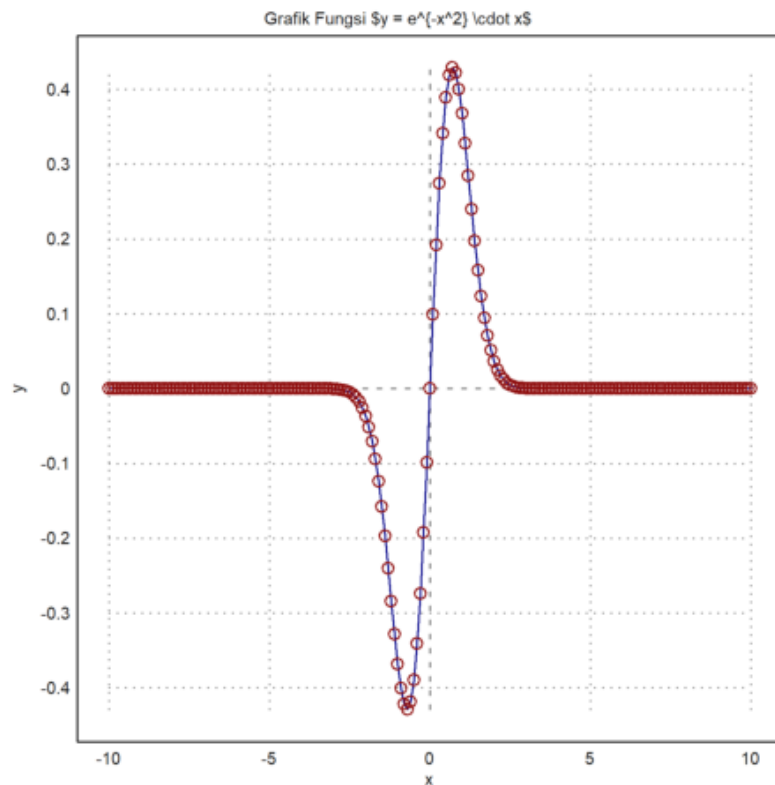
Gambarlah grafik dari data  $y=(e^{-x})^2 \cdot x$  untuk nilai  $x$  dari -10 hingga 10. Pastikan grafik tersebut menunjukkan kurva dengan titik-titiknya.

---

**Penyelesaian :**

---

```
>x = -10:0.1:10;
>y = exp(-x^2) * x;
>plot2d(x, y, style="-", color=blue); // plot the curve
>plot2d(x, y, points=true, add=true, style="o", color=red); // add points
>title("Grafik Fungsi $y = e^{-x^2} \cdot x$");
>xlabel("x");
>ylabel("y");
```




---

2

---

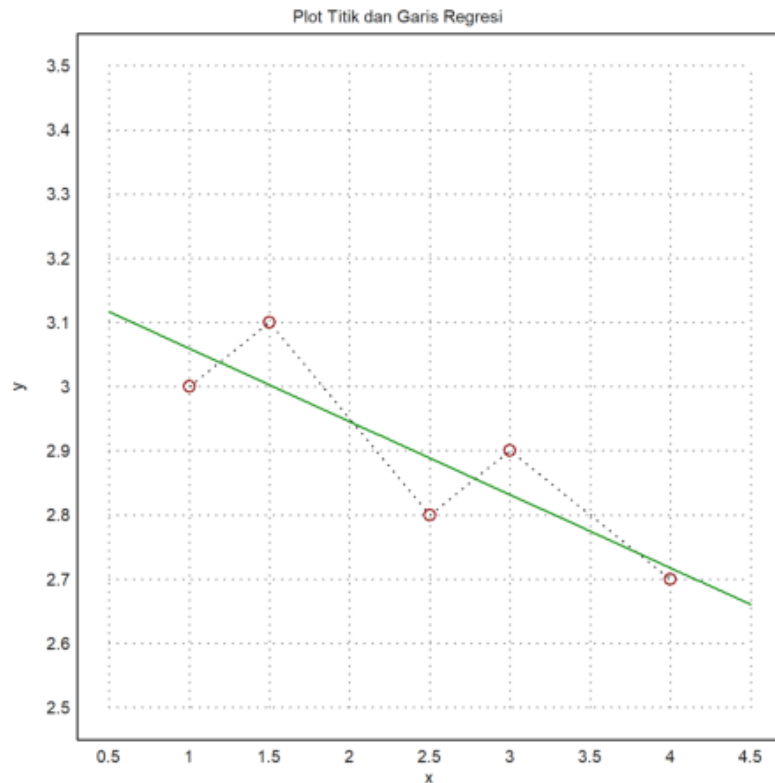
Gambarlah grafik menggunakan data titik berikut:  
 (1,3),(1.5,3.1),(2.5,2.8),(3,2.9),(4,2.7). Tambahkan garis regresi linear ke plot.

---

**Penyelesaian :**

---

```
>xdata = [1, 1.5, 2.5, 3, 4];
>ydata = [3, 3.1, 2.8, 2.9, 2.7];
>plot2d(xdata, ydata, a=0.5, b=4.5, c=2.5, d=3.5, style="."); // plot lines
>plot2d(xdata, ydata, points=true, add=true, style="o", color=red); // add points
>p = polyfit(xdata, ydata, 1); // get regression line
>plot2d("polyval(p, x)", add=true, color=green); // add plot of line
>title("Plot Titik dan Garis Regresi");
>xlabel("x");
>ylabel("y");
```




---

## Menggambar Daerah Yang Dibatasi Kurva

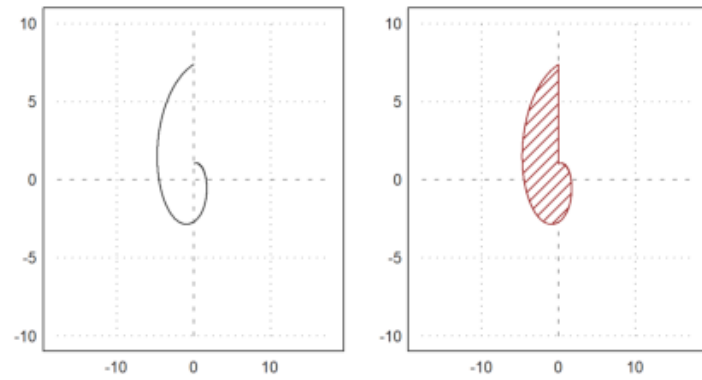
---

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.

- terisi=benar mengisi plot.
- style="...": Pilih dari "", "/", "\", "\/".
- fillcolor: Lihat di atas untuk warna yang tersedia.

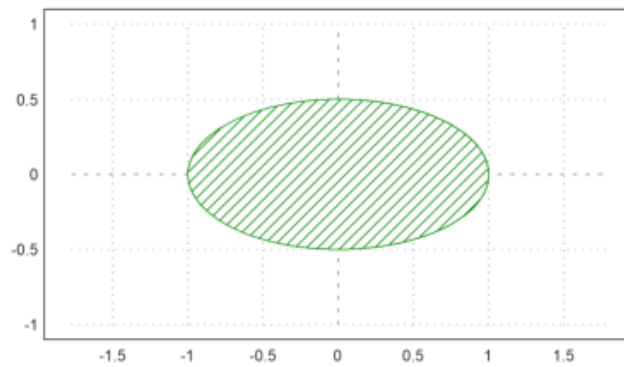
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```

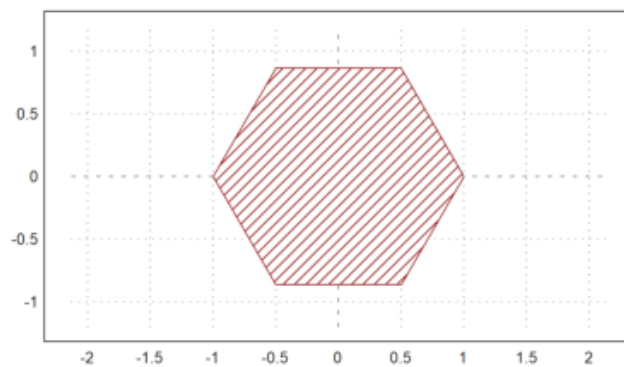


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```

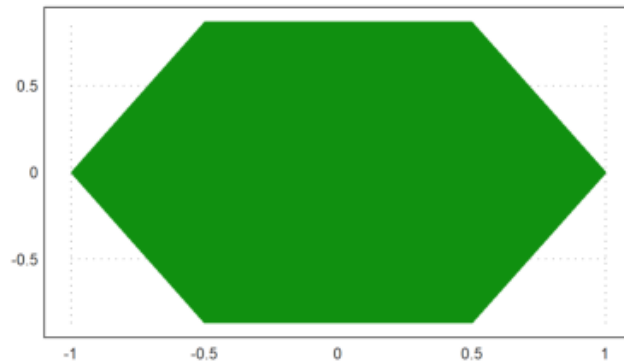


```
>t=linspace(0,2pi,6); ...  
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```



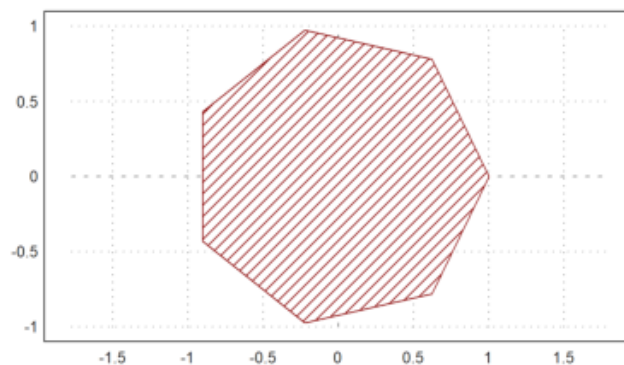


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



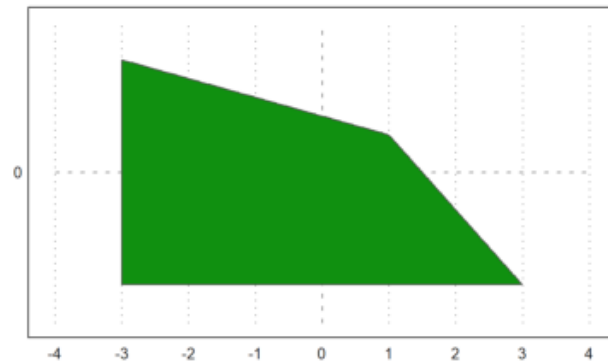
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>t=linspace(0,2pi,7); ...  
> plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=red):
```



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah  $A[k].v \leq 3$  untuk semua baris  $A$ . Untuk mendapatkan sudut yang bagus, kita menggunakan  $n$  yang relatif besar.

```
>A=[2,1;1,2;-1,0;0,-1];  
>function f(x,y) := max([x,y].A');  
>plot2d("f",r=4,level=[0;3],color=green,n=111):
```



Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

Kami sekarang memiliki vektor  $x$  dan  $y$  nilai. `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk isi.

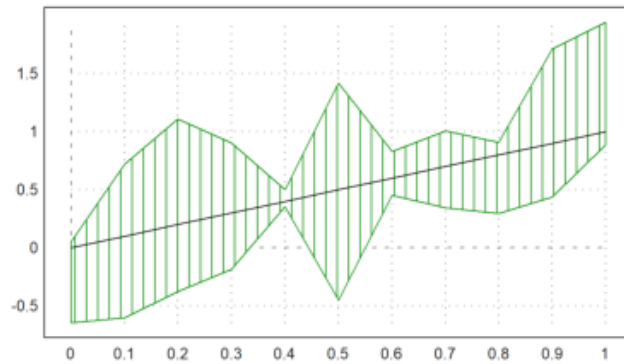
```
>plot2d(x,y,<grid,<frame,>filled):
```



Sebuah vektor interval diplot terhadap nilai  $x$  sebagai daerah terisi antara nilai interval bawah dan atas.

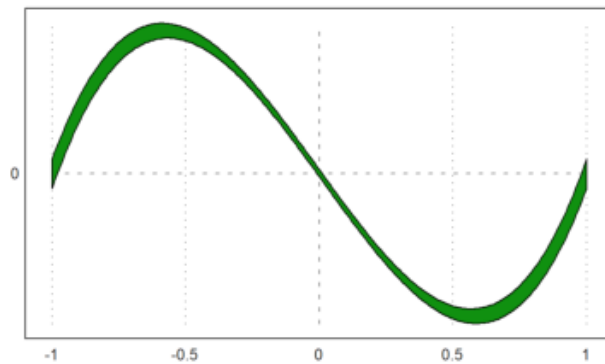
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

```
>t=0:0.1:1; ...
> plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|"); ...
> plot2d(t,t,add=true):
```



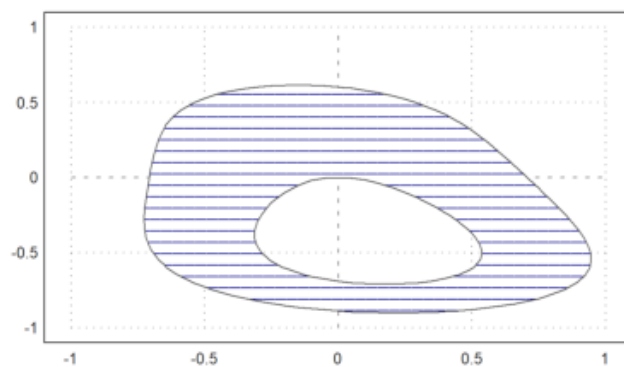
Jika  $x$  adalah vektor yang diurutkan, dan  $y$  adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
```



Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks  $2 \times n$ . Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

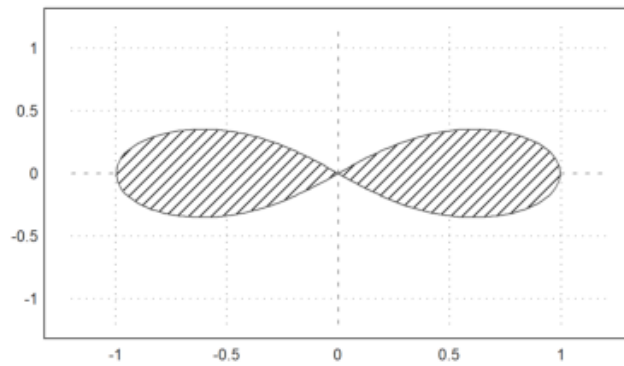
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



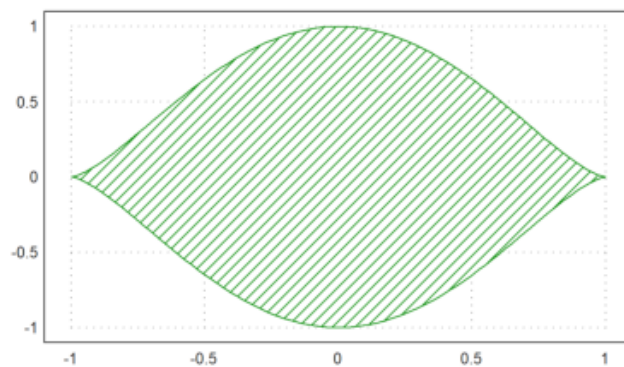
Kami juga dapat mengisi rentang nilai seperti

$$-1 \leq (x^2 + y^2)^2 - x^2 + y^2 \leq 0.$$

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/") :
```



```
>plot2d("cos(x)", "sin(x)^3",xmin=0,xmax=2pi,>filled,style="/") :
```



---

Catatan Baru

---

contoh penyelesaian masalah

---

1

---

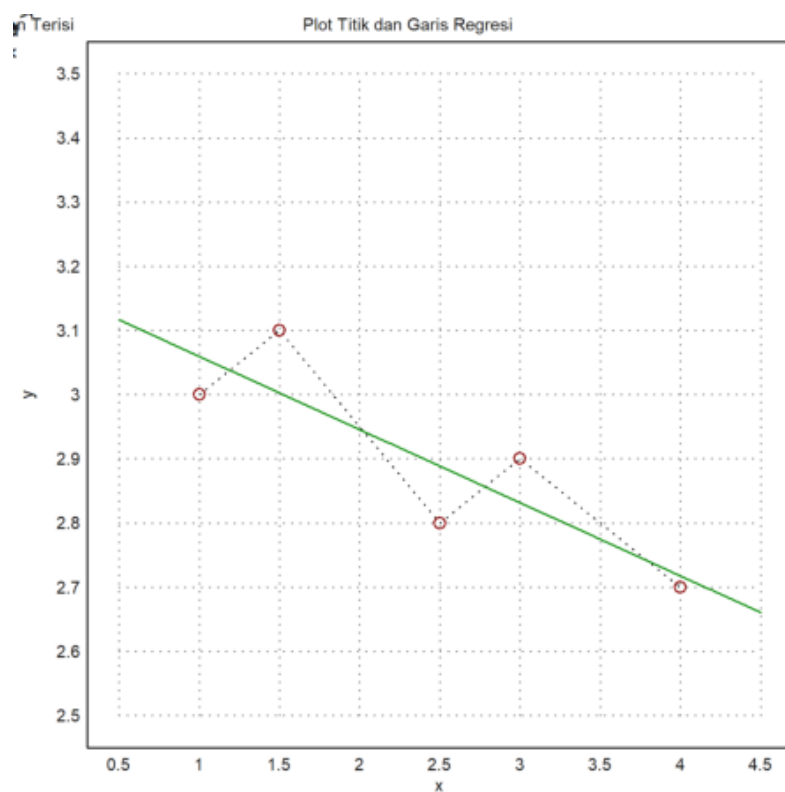
Gambarlah daerah yang dibatasi oleh kurva lingkaran dengan jari-jari 1 yang terisi dengan warna merah.

---

Penyelesaian :

---

```
>t = linspace(0, 2*pi, 1000); // parameter untuk kurva
>x = cos(t); // x(t)
>y = sin(t); // y(t)
>figure(1);
>plot2d(x, y, filled=true, style="/", fillcolor=red); // mengisi lingkaran
>title("Lingkaran Terisi");
>xlabel("x");
>ylabel("y");
```



---

2

---

Gambarlah daerah yang dibatasi oleh fungsi  $y=\sin(x)$  dan  $y=\cos(x)$  pada interval  $[0, \pi/2]$  dengan warna biru.

---

Penyelesaian :

---

```

>x = linspace(0, pi/2, 1000);
>y1 = sin(x); // sin(x)
>y2 = cos(x); // cos(x)
>figure(2);
>plot2d(x, y1, filled=true, style="/", fillcolor=blue, level=[0; 1]); // mengisi antara si

```

Contour needs a real matrix!

datacontour:

```
contour(Z,level);
```

Try "trace errors" to inspect local variables after errors.

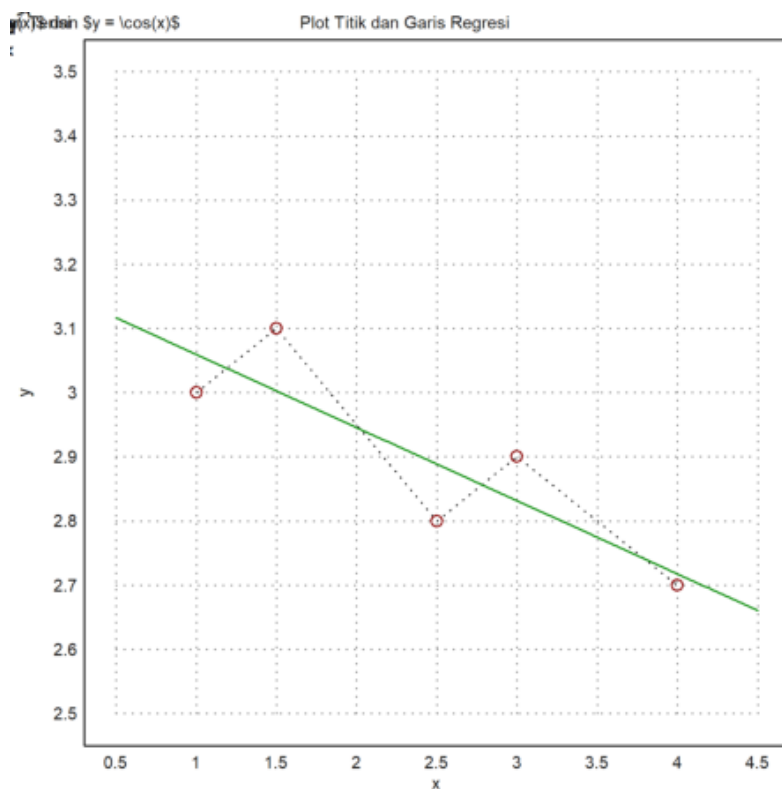
plot2d:

```
=style,=outline,=frame);
```

```

>plot2d(x, y2, add=true, style="-", color=orange); // menambahkan cos(x)
>title("Daerah antara $y = \sin(x)$ dan $y = \cos(x)$");
>xlabel("x");
>ylabel("y"):

```



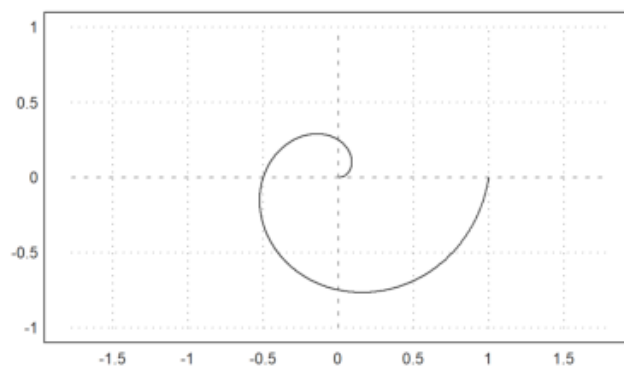
Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.

Dalam contoh berikut, kami memplot spiral

$$\gamma(t) = t \cdot (\cos(2\pi t), \sin(2\pi t))$$

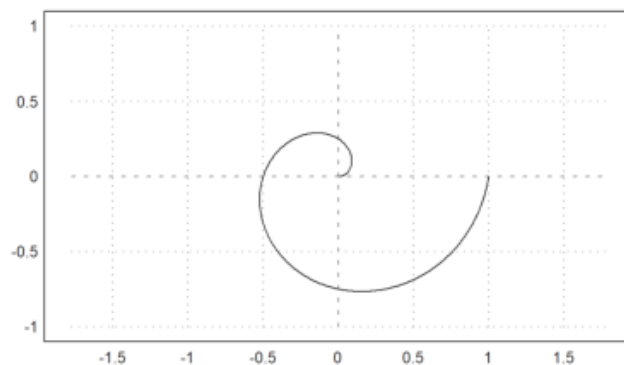
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

```
>t=linspace(0,1,1000); ...  
>plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```

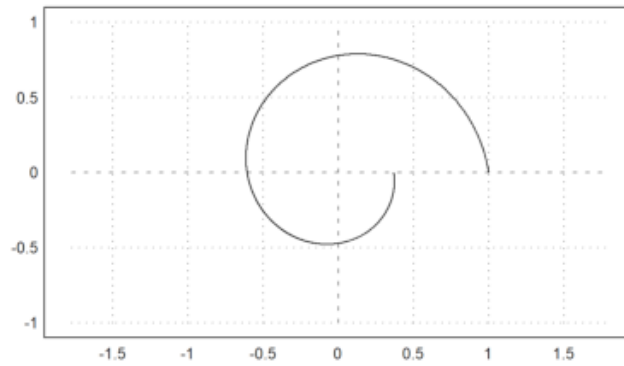


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
>plot2d("x*cos(2*pi*x)", "x*sin(2*pi*x)", xmin=0, xmax=1, r=1):
```



```
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2*pi*t); y=r*sin(2*pi*t);  
>plot2d(x,y,r=1):
```



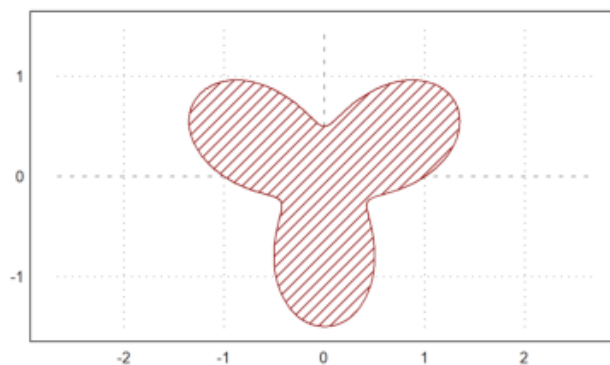
Dalam contoh berikutnya, kami memplot kurva

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



## Menggambar Grafik Bilangan Kompleks

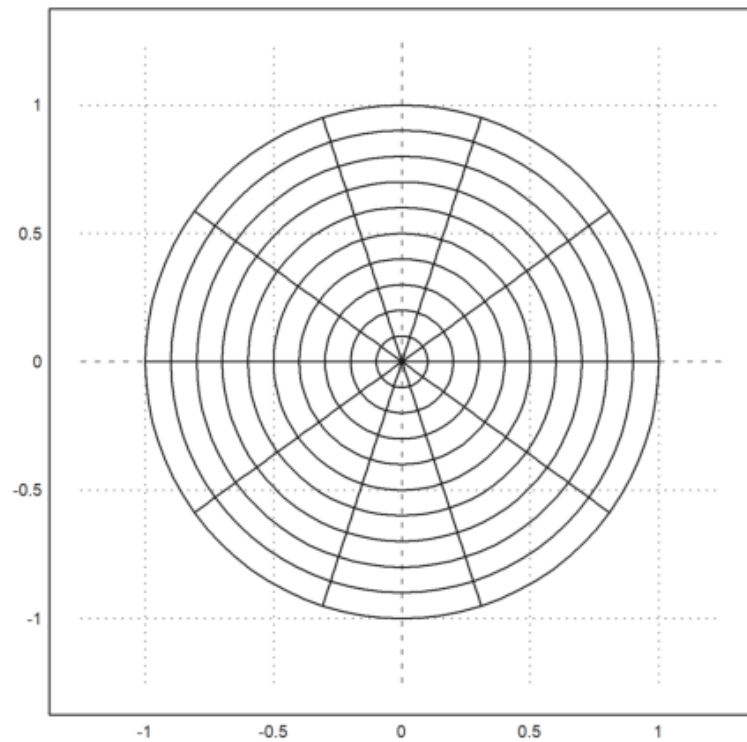
Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

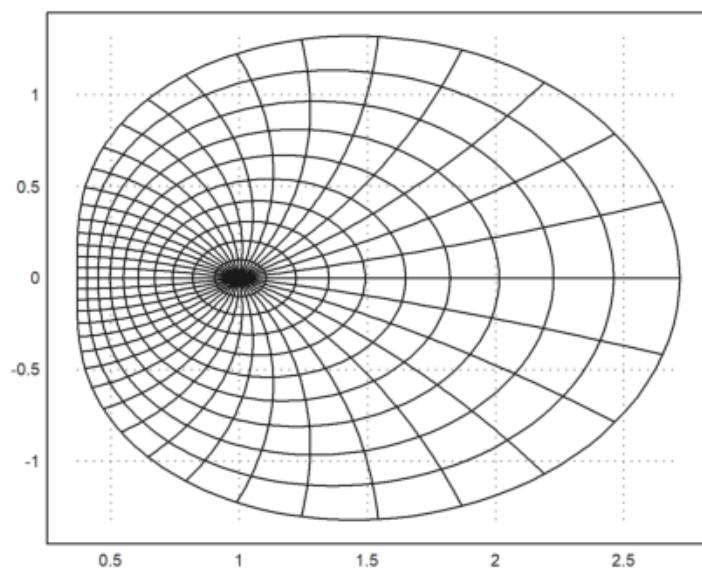
Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```

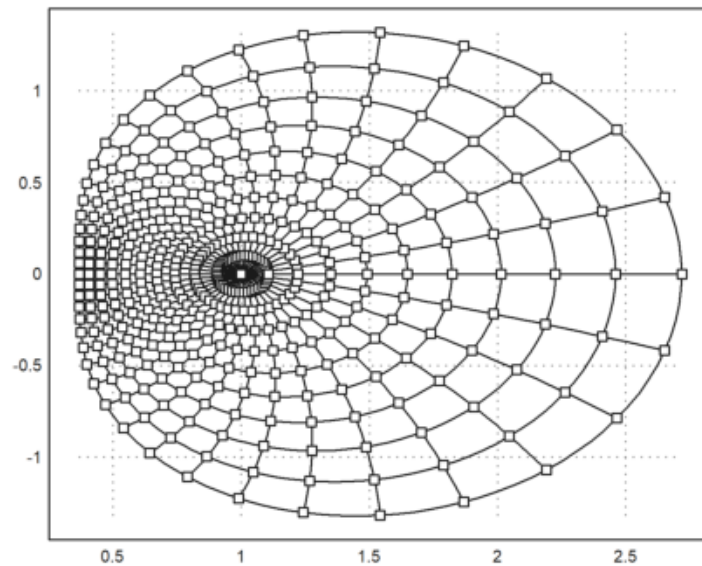




```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```



```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

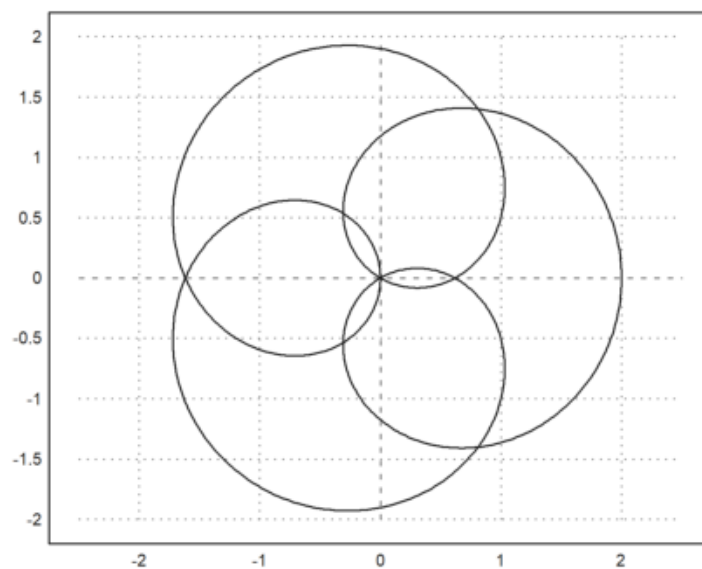


Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh, kami memplot lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

```
>t=linspace(0,2pi,1000); ...  
>plot2d(exp(I*t)+exp(4*I*t),r=2):
```



Ada banyak fungsi yang dikhususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

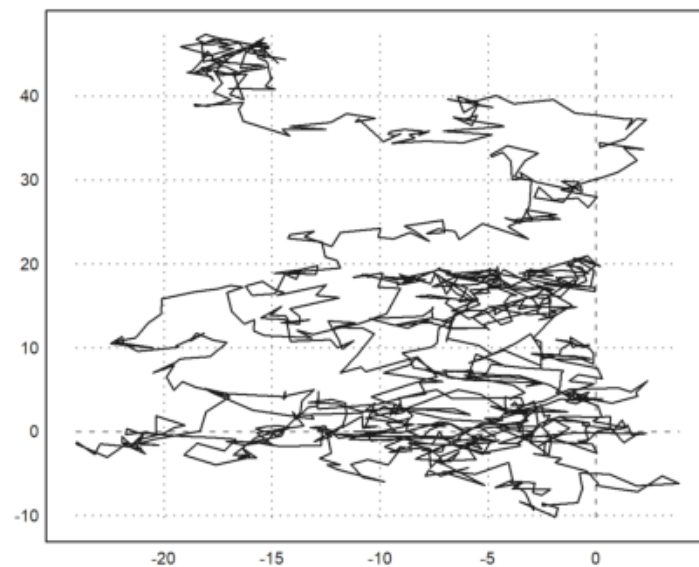
Jumlah kumulatif dari nilai terdistribusi 0-1-normal menghasilkan jalan acak.

```
>plot2d(cumsum(randnormal(1,1000))) :
```

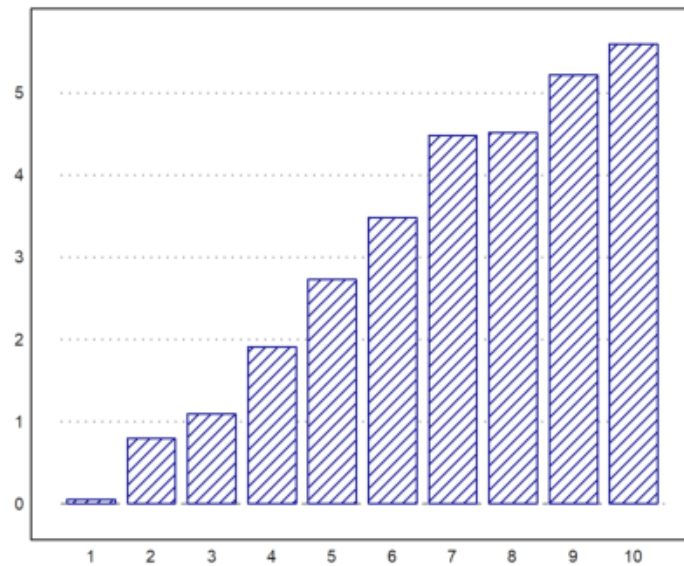


Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]) :
```

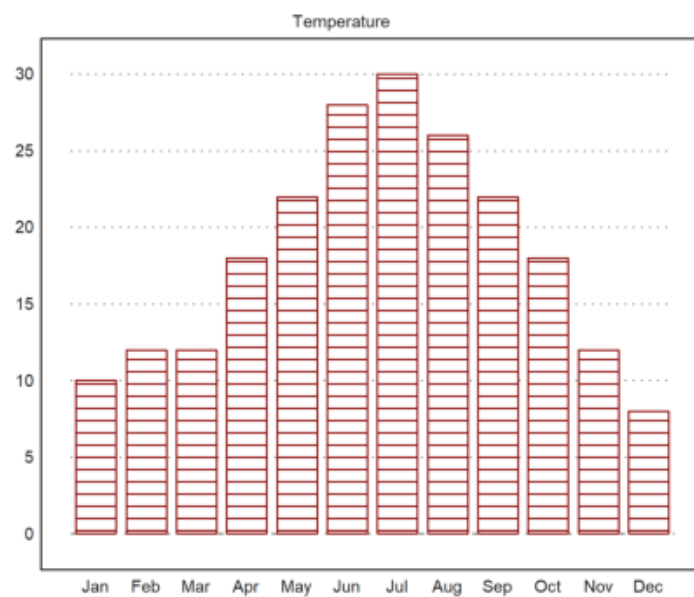


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

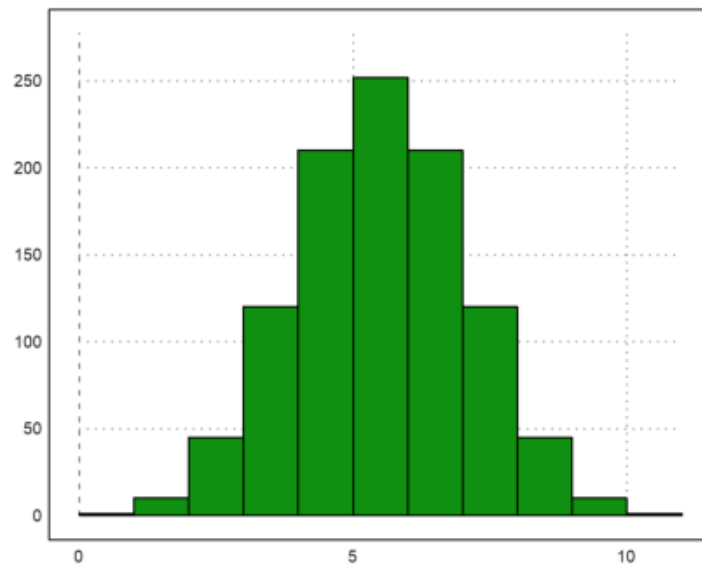


Itu juga dapat menampilkan string sebagai label.

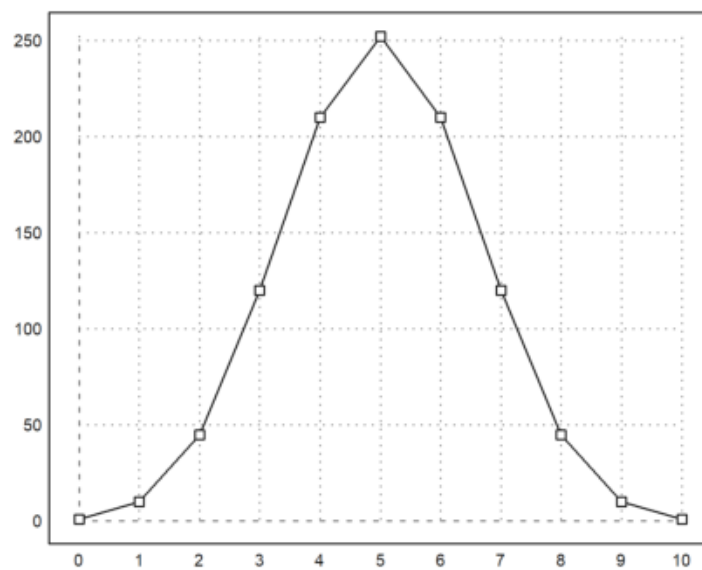
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...  
> "Jul","Aug","Sep","Oct","Nov","Dec"];  
>values=[10,12,12,18,22,28,30,26,22,18,12,8];  
>columnsplot(values,lab=months,color=red,style="-");  
>title("Temperature"):
```



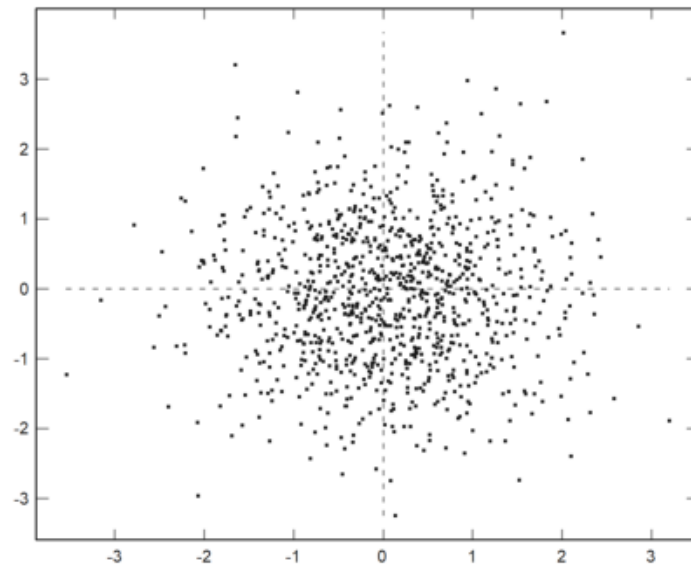
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



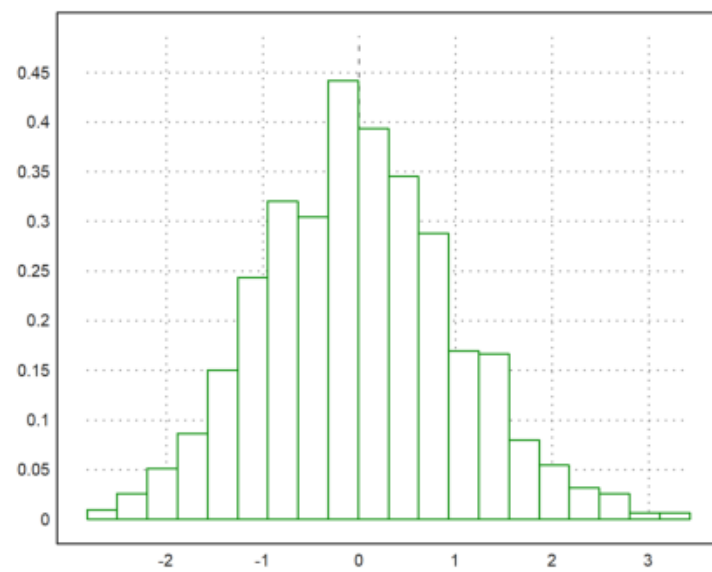
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



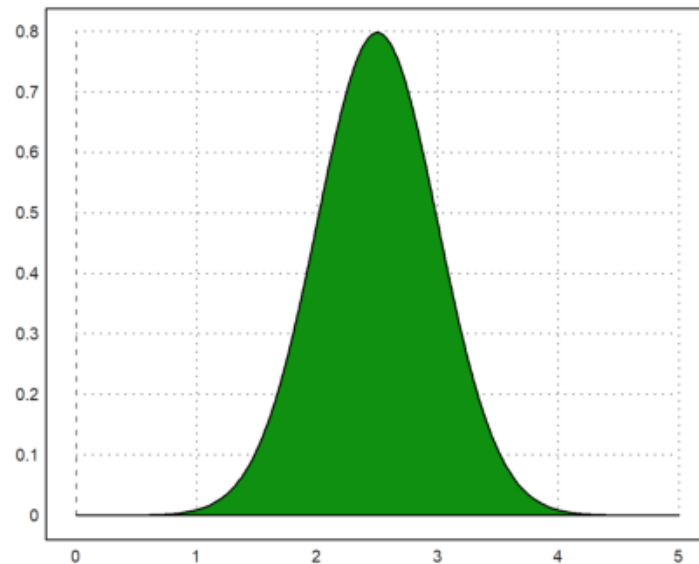
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



```
>plot2d(normal(1,1000),>distribution,style="O"):
```

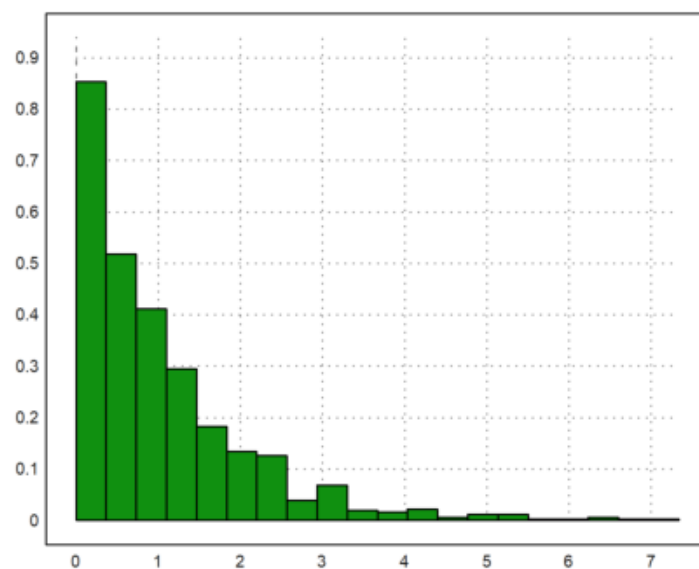


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



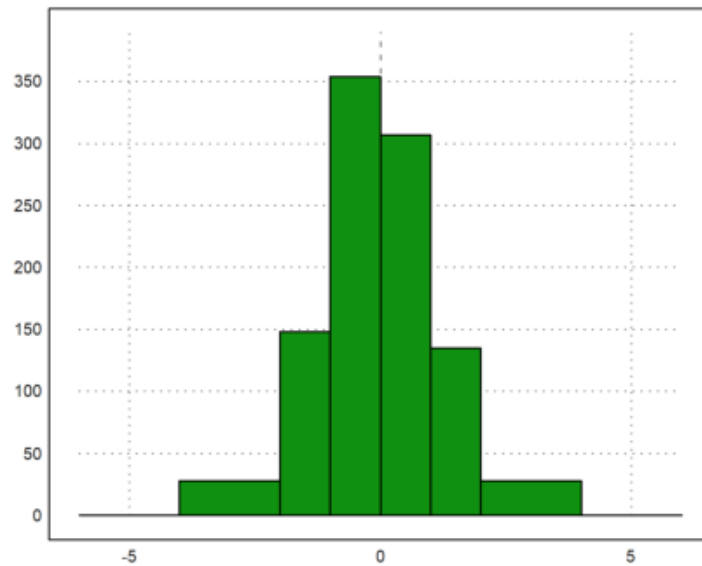
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan `distribution=n` dengan `plot2d`.

```
>w=randexponential(1,1000); // exponential distribution
>plot2d(w,>distribution): // or distribution=n with n intervals
```



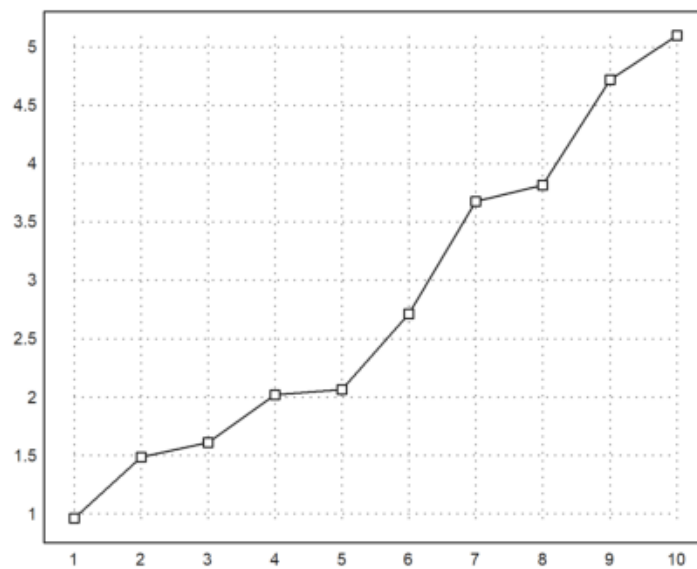
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan `>bar` di `plot3d`, atau dengan `plot` kolom.

```
>w=normal(1000); // 0-1-normal distribution
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // interval bounds v
>plot2d(x,y,>bar):
```



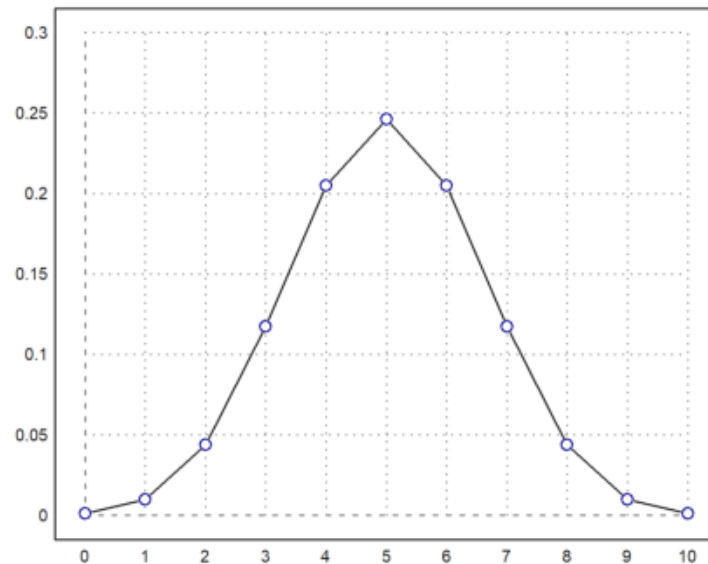
Fungsi `statplot()` menyetel gaya dengan string sederhana.

```
>statplot(1:10,cumsum(random(10)),"b"):
```



```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```

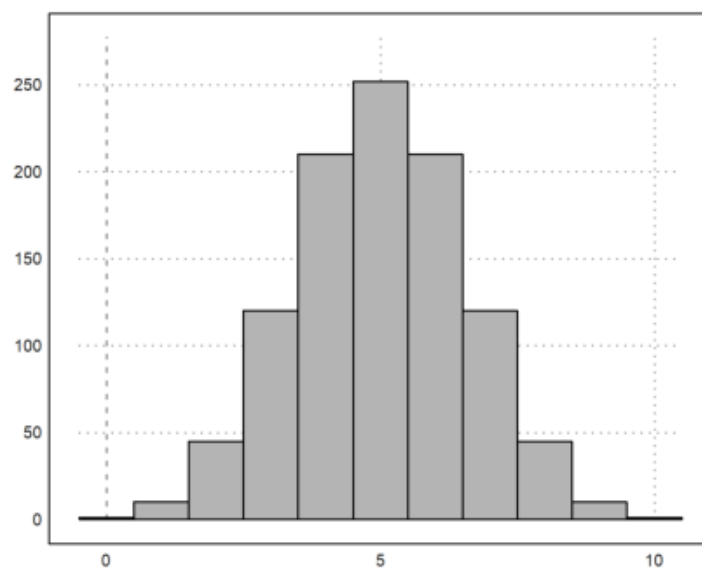




Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Bilah akan memanjang dari  $x[i]$  ke  $x[i+1]$  dengan nilai  $y[i]$ . Jika x memiliki ukuran yang sama dengan y, maka akan diperpanjang satu elemen dengan spasi terakhir.

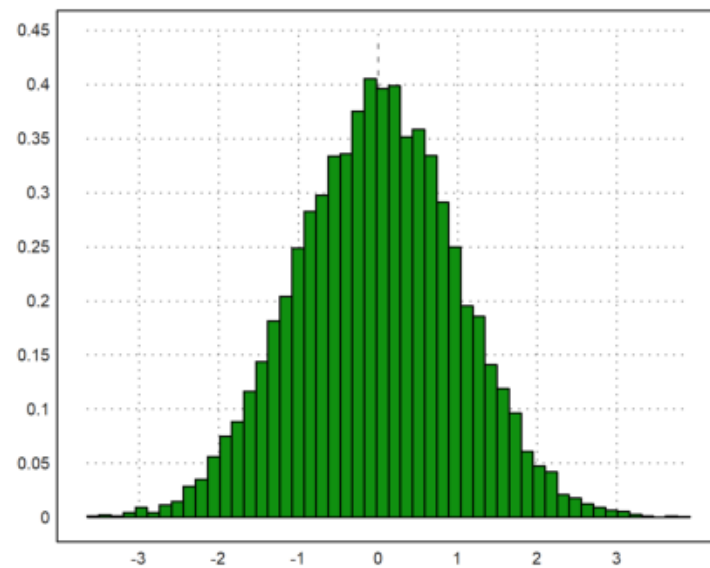
Gaya isian dapat digunakan seperti di atas.

```
>n=10; k=bin(n,0:n); ...
>plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
```

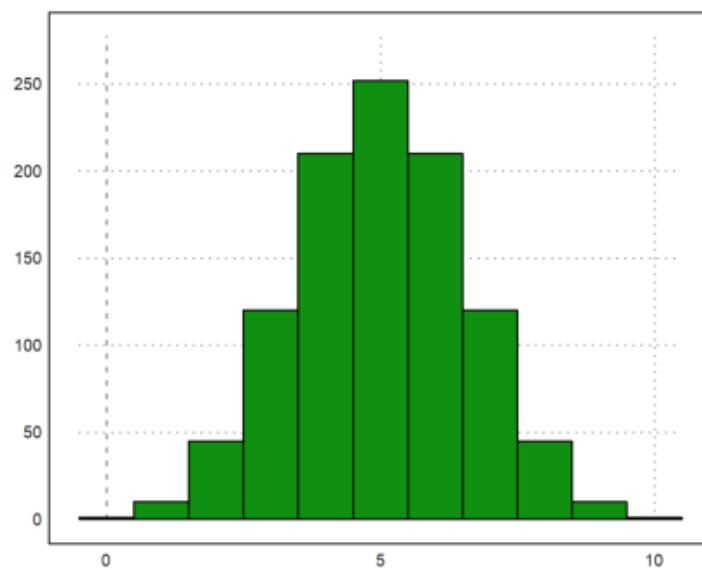


Data untuk plot batang (`bar=1`) dan histogram (`histogram=1`) dapat dinyatakan secara eksplisit dalam `xv` dan `yv`, atau dapat dihitung dari distribusi empiris dalam `xv` dengan `>distribusi` (atau `distribusi=n`). Histogram nilai `xv` akan dihitung secara otomatis dengan `>histogram`. Jika `>genap` ditentukan, nilai `xv` akan dihitung dalam interval bilangan bulat.

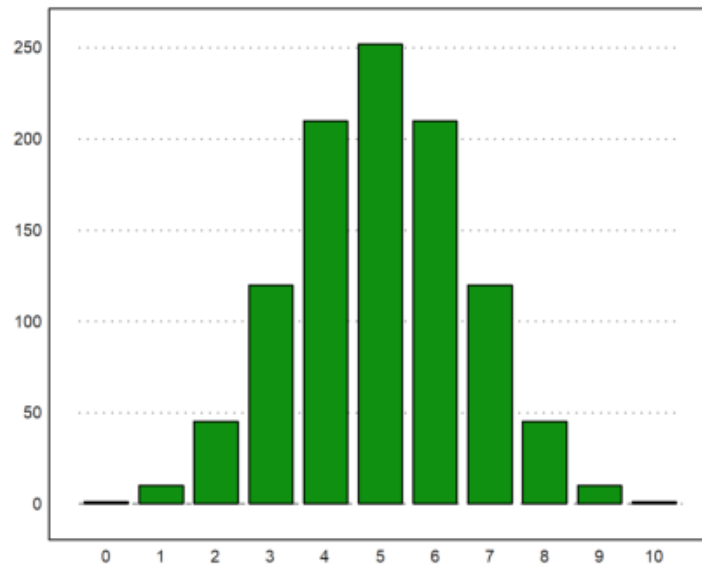
```
>plot2d(normal(10000),distribution=50):
```



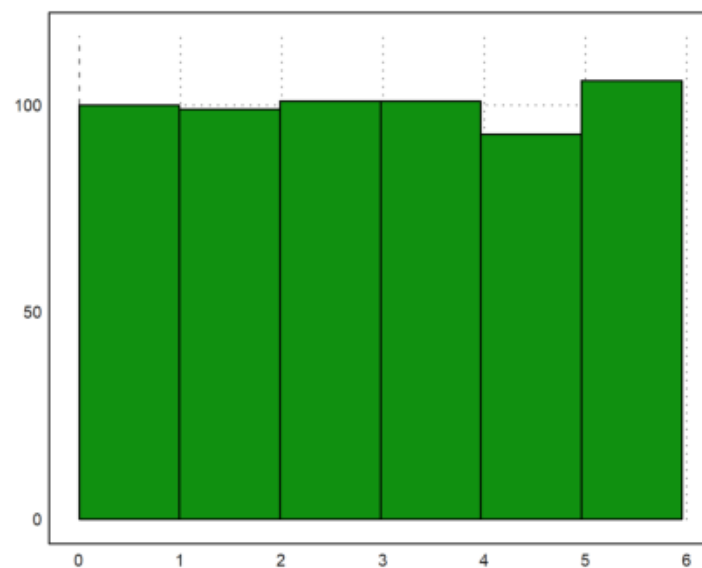
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



```
>columnsplo(m,k):
```

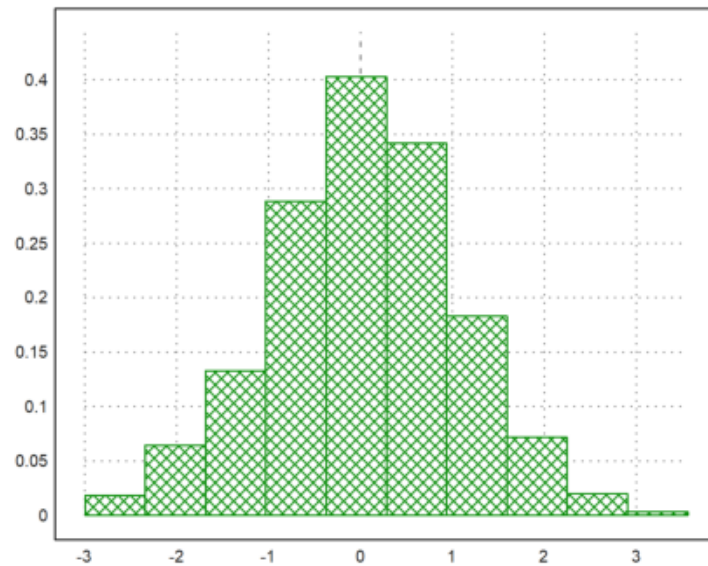


```
>plot2d(random(600)*6,histogram=6):
```



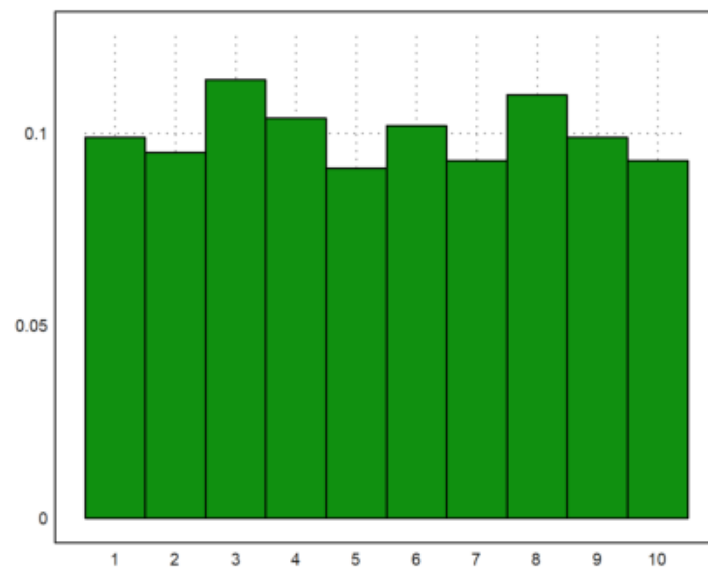
Untuk distribusi, ada parameter distribusi=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

```
>plot2d(normal(1,1000),distribution=10,style="\/"): 
```



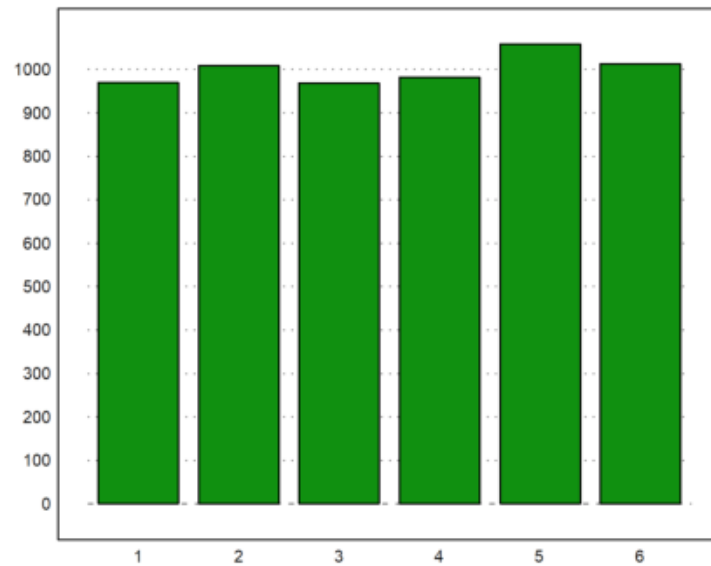
Dengan parameter `even=true`, ini akan menggunakan interval integer.

```
>plot2d(inrandom(1,1000,10),distribution=10,even=true):
```

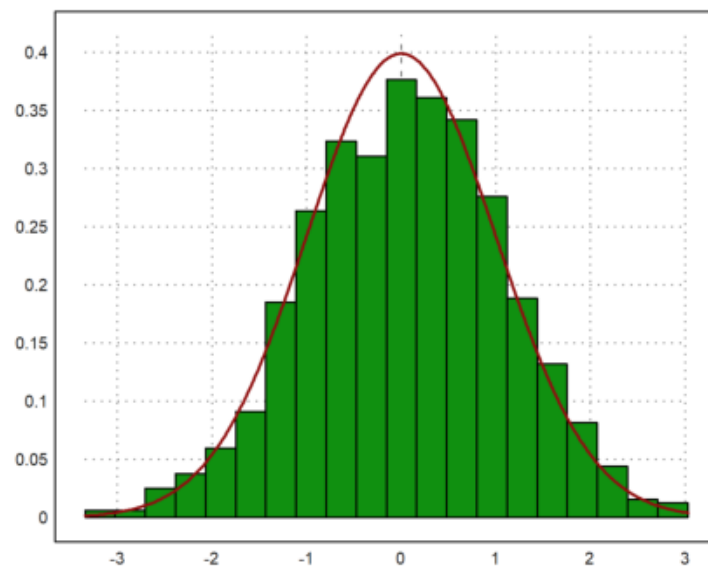


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

```
>columnsplot(getmultiplicities(1:6,inrandom(1,6000,6))):
```

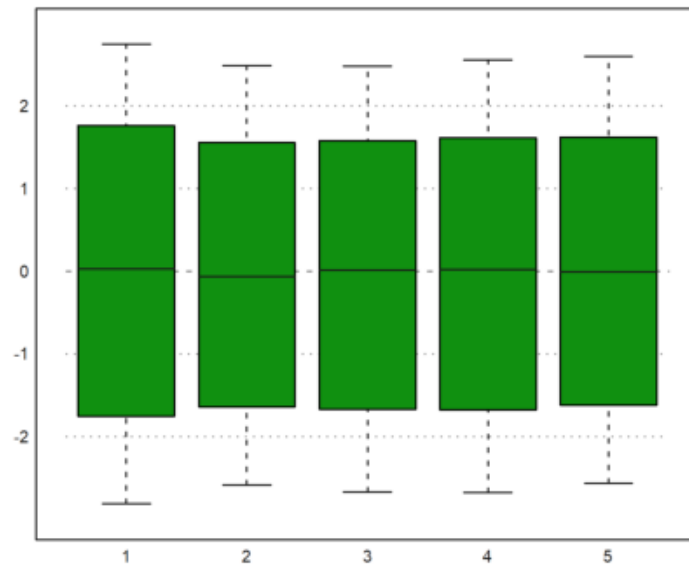


```
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak plot khusus untuk statistik. Sebuah boxplot menunjukkan kuartil dari distribusi ini dan banyak dari outlier. Menurut definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali kisaran 50% tengah plot.

```
>M=normal(5,1000); boxplot(quartiles(M)):
```



## Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan  $f(x,y)=\text{level}$ , di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika  $\text{level}=\text{"auto"}$ , akan ada garis level  $n_c$ , yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan  $>\text{hue}$  untuk menunjukkan nilai fungsi. Untuk fungsi implisit,  $xv$  harus berupa fungsi atau ekspresi dari parameter  $x$  dan  $y$ , atau, sebagai alternatif,  $xv$  dapat berupa matriks nilai.

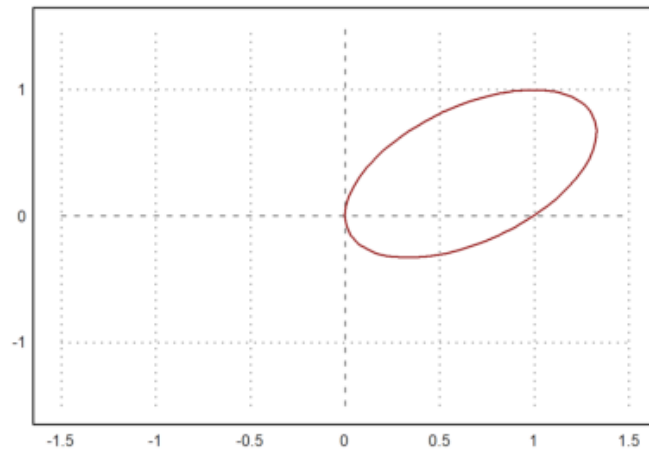
Euler dapat menandai garis level

$$f(x,y) = c$$

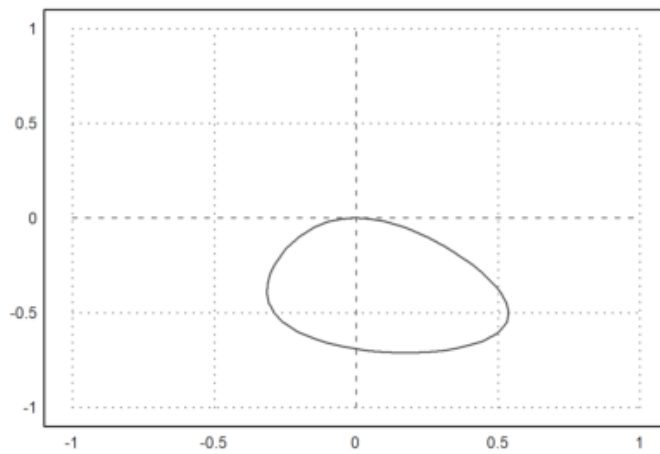
dari fungsi apapun.

Untuk menggambar himpunan  $f(x,y)=c$  untuk satu atau lebih konstanta  $c$ , Anda dapat menggunakan `plot2d()` dengan plot implisitnya di dalam bidang. Parameter untuk  $c$  adalah  $\text{level}=c$ , di mana  $c$  dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

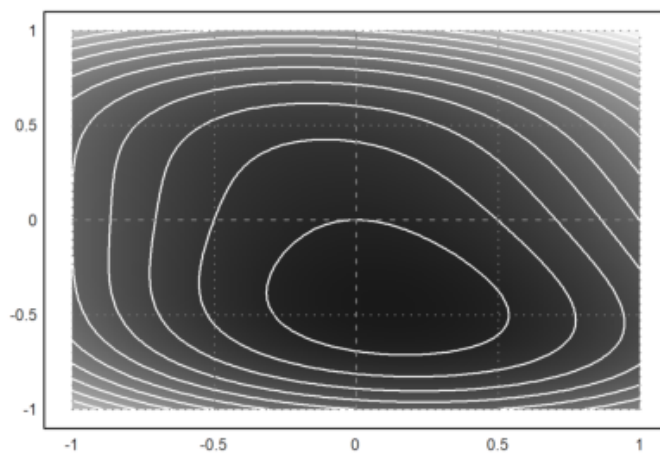
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



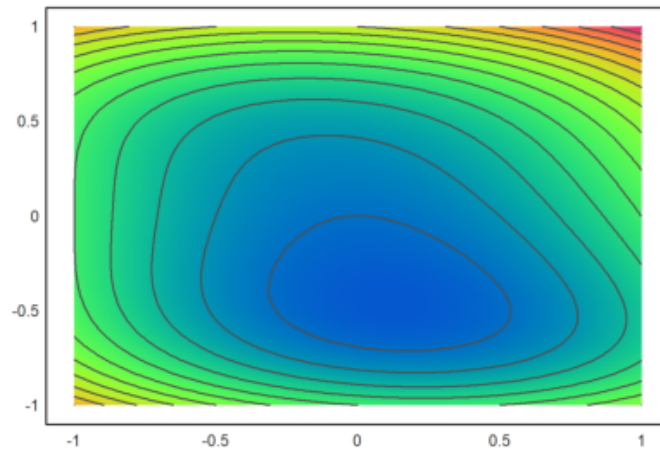
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```



```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
```

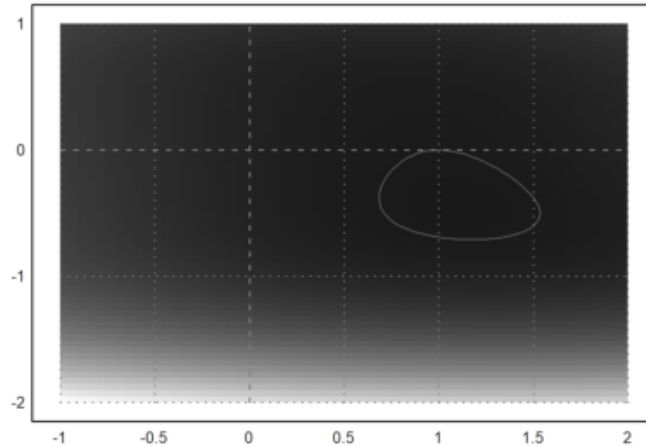


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```



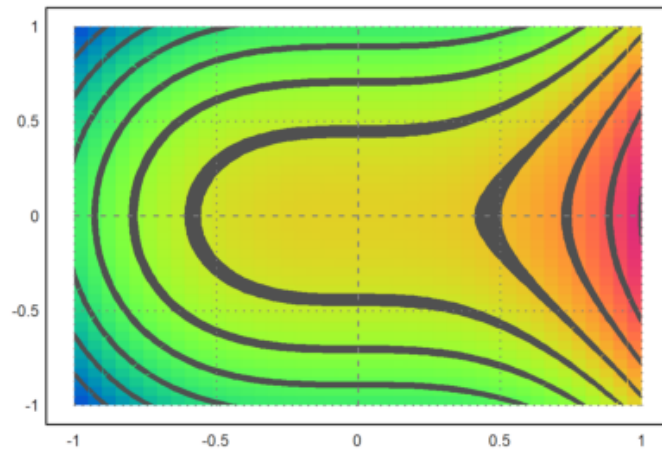
Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```

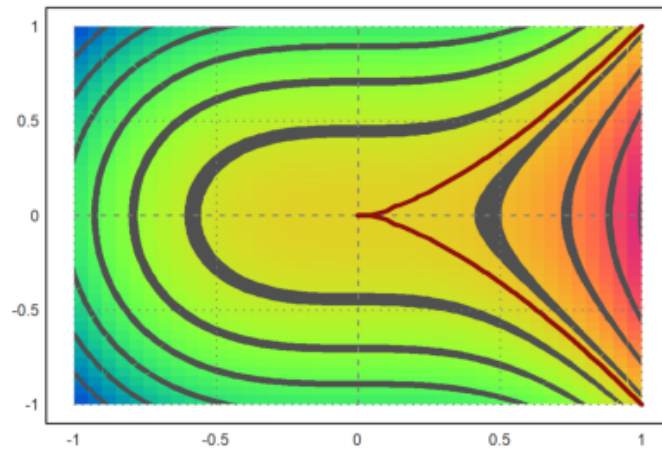


```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```

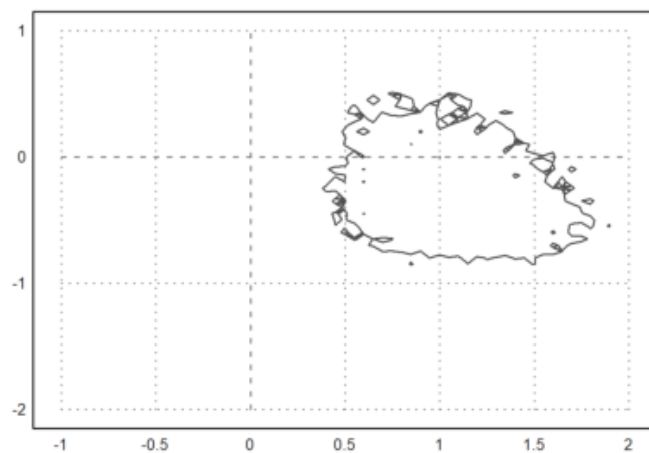




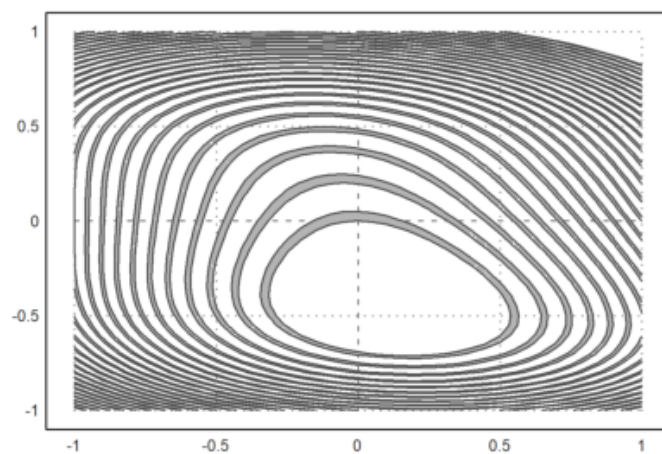
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



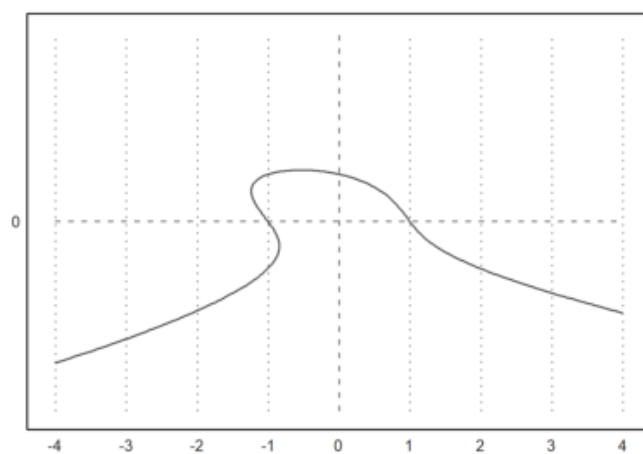
```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



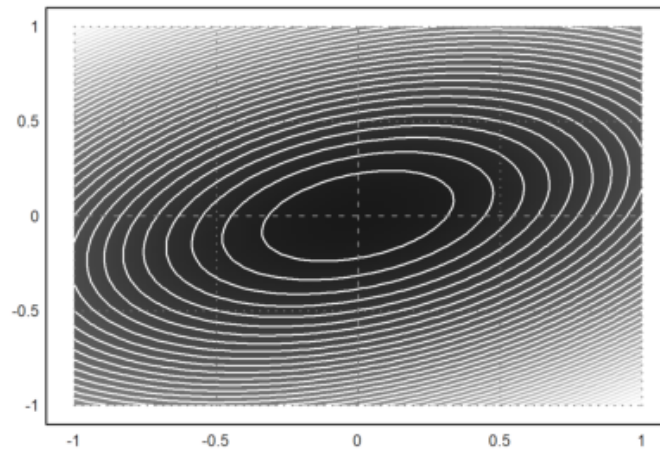
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



```
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
```



```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```



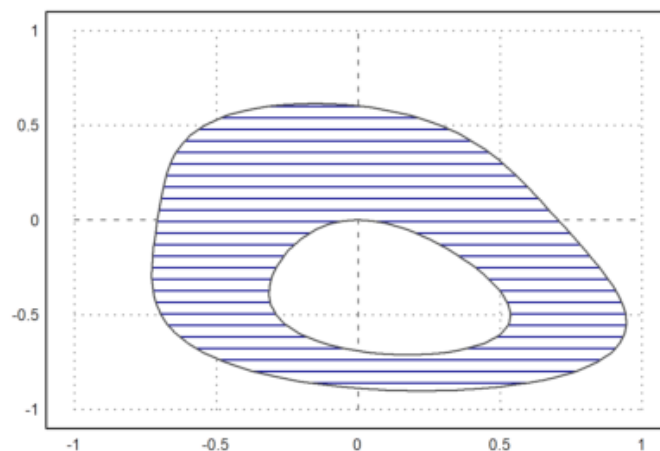
Juga dimungkinkan untuk mengisi set

$$a \leq f(x, y) \leq b$$

dengan rentang tingkat.

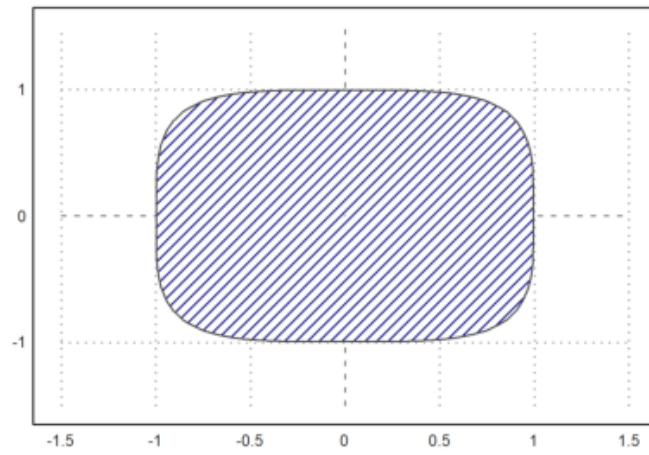
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>plot2d(expr, level=[0;1], style="-", color=blue): // 0 <= f(x, y) <= 1
```

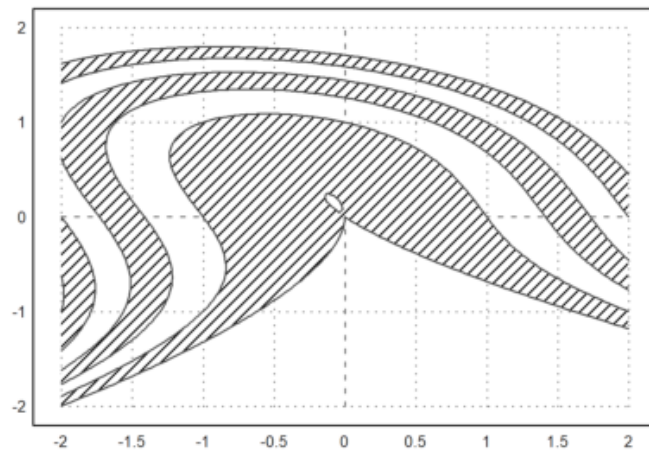


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

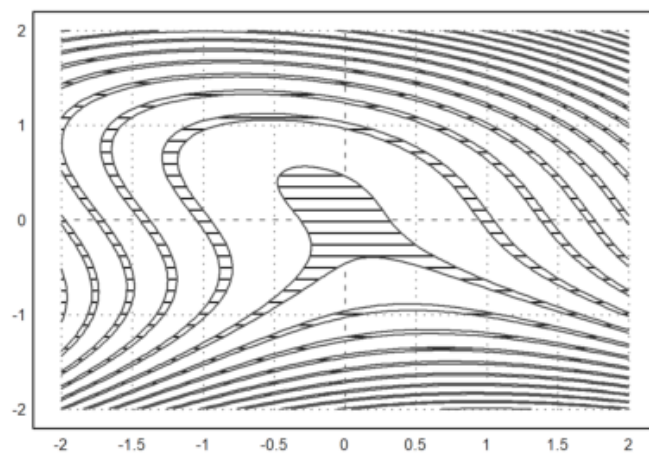
```
>plot2d("x^4+y^4", r=1.5, level=[0;1], color=blue, style="/"):
```



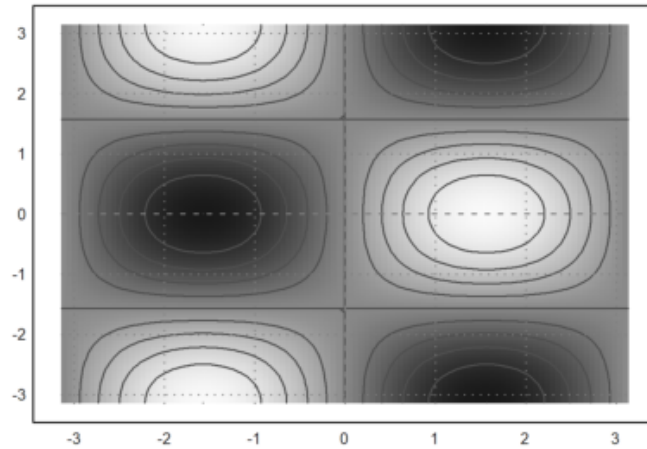
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```



```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

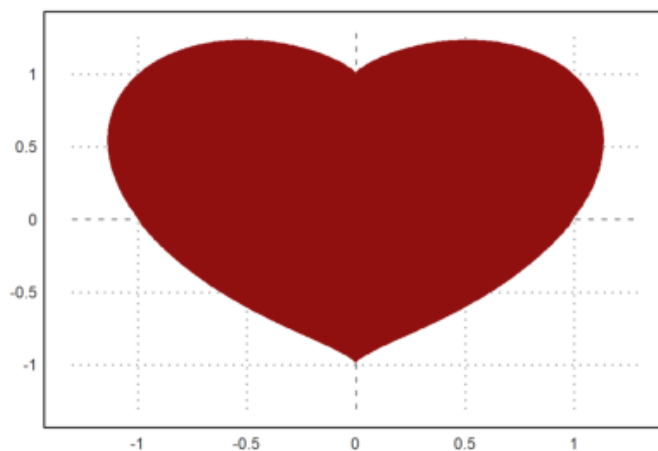


Dimungkinkan juga untuk menandai suatu wilayah

$$a \leq f(x,y) \leq b.$$

Ini dilakukan dengan menambahkan level dengan dua baris.

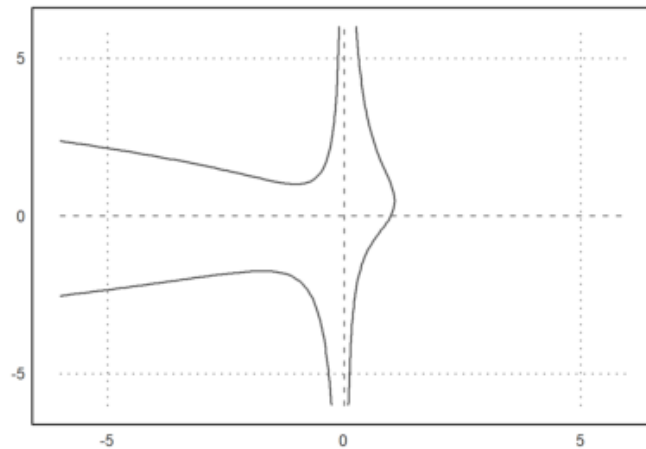
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
> style="#",color=red,<outline, ...
> level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi persamaan seperti

$$x^3 - xy + x^2y^2 = 6$$

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
```



```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```

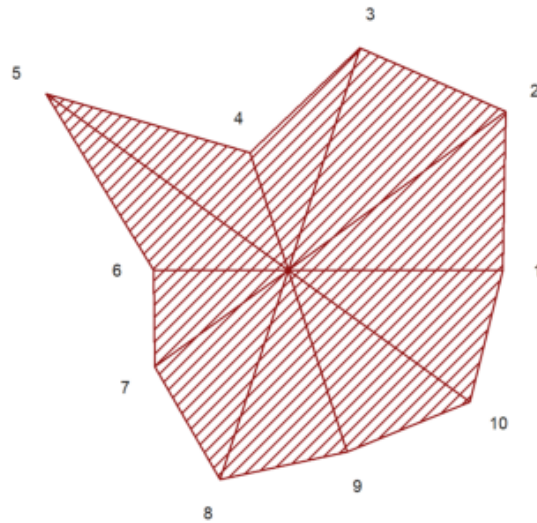
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
  if lab!=none then
    rlab=v[#]+r*0.1;
    {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
    ctext(""+lab[#],col,row-textheight()/2);
  endif;
end;
barcolor(cl); barstyle(st);
holding(h);
window(w);
endfunction

```

Tidak ada kotak atau sumbu kutu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin plot Anda berhasil.

```
>reset; starplot1(normal(1,10)+5,color=red,lab=1:10):
```



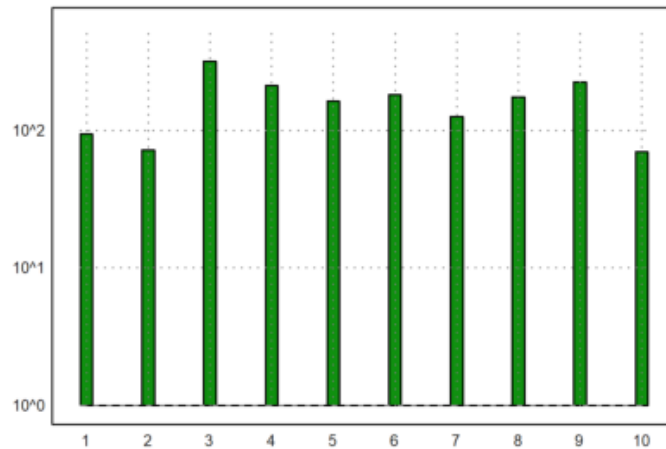
Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir. Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```
>function logimpulseplot1 (x,y) ...

    {x0,y0}=makeimpulse(x,log(y)/log(10));
    plot2d(x0,y0,>bar,grid=0);
    h=holding(1);
    frame();
    xgrid(ticks(x));
    p=plot();
    for i=-10 to 10;
        if i<=p[4] and i>=p[3] then
            ygrid(i,yt="10^"+i);
        endif;
    end;
    holding(h);
endfunction
```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

```
>aspect(1.5); x=1:10; y=-log(random(size(x)))*200; ...
>logimpulseplot1(x,y):
```



Mari kita menganimasikan kurva 2D menggunakan plot langsung. Perintah `plot(x,y)` hanya memplot kurva ke jendela plot. `setplot(a,b,c,d)` mengatur jendela ini.

Fungsi `wait(0)` memaksa plot untuk muncul di jendela grafik. Jika tidak, menggambar ulang terjadi dalam interval waktu yang jarang.

```
>function animliss (n,m) ...

t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=linewidth(2);
setplot(-1,1,-1,1);
repeat
    clg;
    plot(sin(n*t),cos(m*t+f));
    wait(0);
    if testkey() then break; endif;
    f=f+0.02;
end;
framecolor(c);
linewidth(l);
endfunction
```

Tekan sembarang tombol untuk menghentikan animasi ini.

```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

## Plot Logaritmik

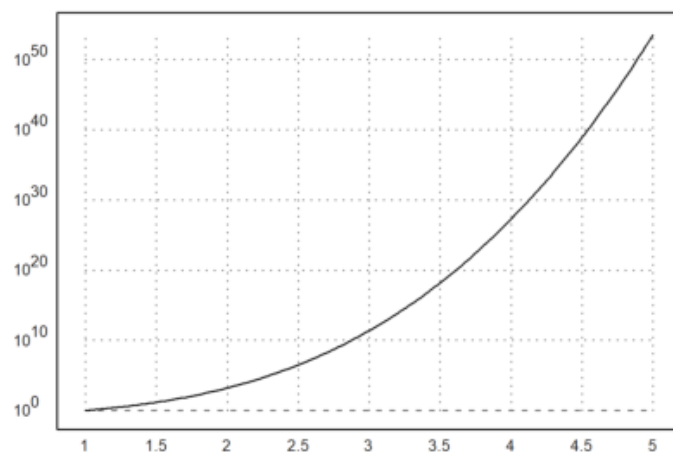
EMT menggunakan parameter "logplot" untuk skala logaritmik.

Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan `logplot=1`, atau menggunakan skala logaritma dalam x dan y dengan `logplot=2`, atau dalam x dengan `logplot=3`.

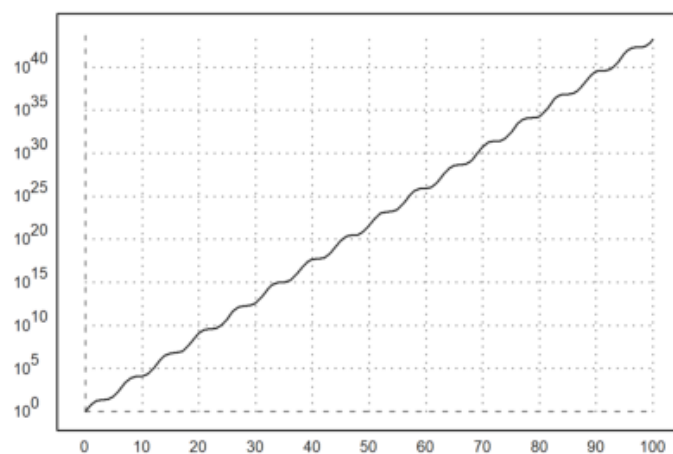


- logplot=1: y-logaritma
- logplot=2: x-y-logaritma
- logplot=3: x-logaritma

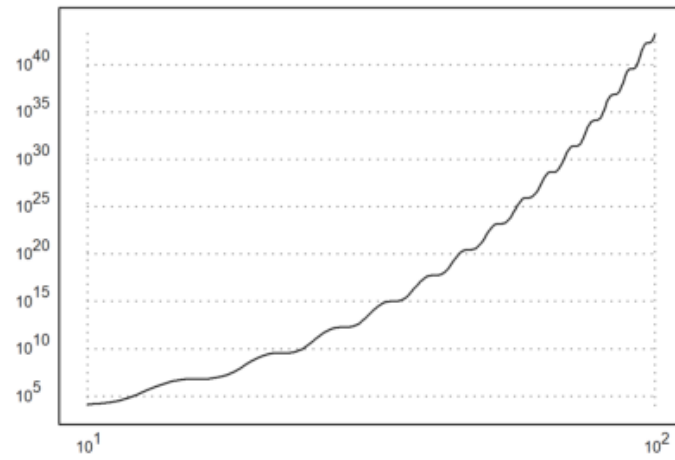
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



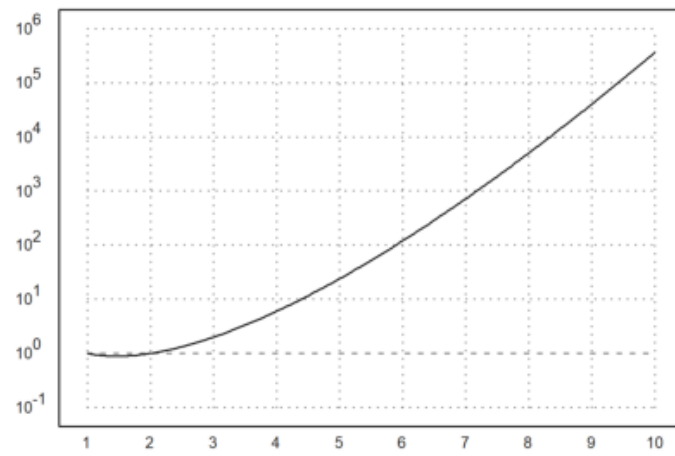
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



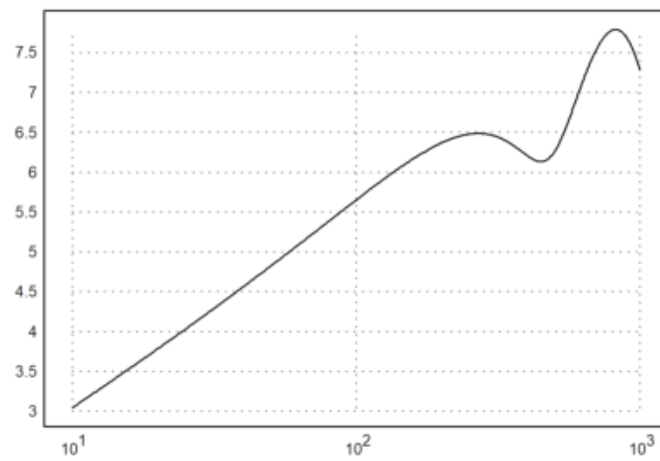
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



```
>plot2d("gamma(x)",1,10,logplot=1):
```

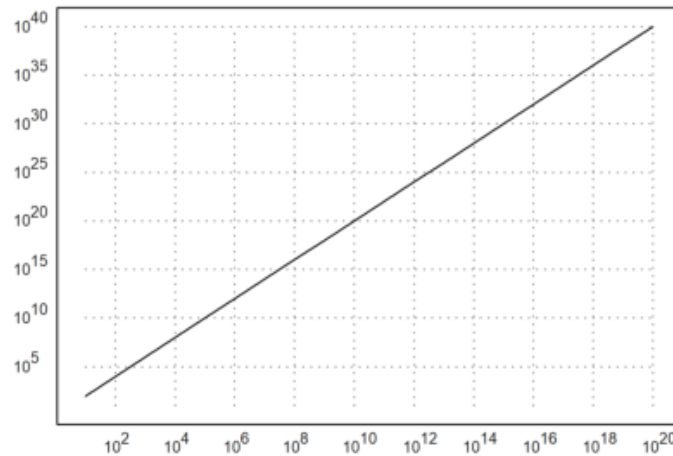


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan plot data.

```
>x=10^(1:20); y=x^2-x;  
>plot2d(x,y,logplot=2):
```



---

## Catatan Baru

---

Dalam EMT, kamu bisa menggunakan parameter logplot untuk menggambar grafik dengan skala logaritmik. Berikut adalah tiga opsi untuk parameter ini:

- a. logplot=1: Menggunakan skala logaritma pada sumbu y(logaritma basis e).
- b. logplot=2: Menggunakan skala logaritma pada kedua sumbu x dan y.
- c. logplot=3: Menggunakan skala logaritma pada sumbu x saja.

## contoh penyelesaian masalah menggambar kurva/plot 2D \*\* 1

### Menggambar

---

Fungsi Eksponensial

$$f(x) = (e^x)^3 - x$$

dengan Skala Logaritma pada Sumbu y

**Penyelesaian :**

---

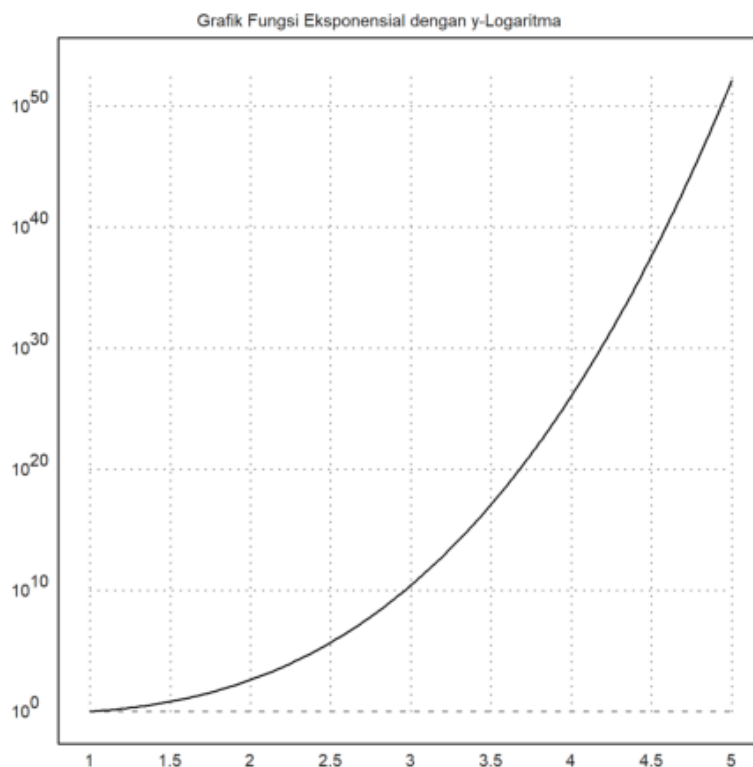
```
>reset; // menghapus semua plot sebelumnya  
>plot2d("exp(x^3 - x)", 1, 5, logplot=1); // menggambar dengan y-logaritma  
>title("Grafik Fungsi Eksponensial dengan y-Logaritma"); // judul  
>xl("x"); // label sumbu x
```

```
Function xl not found.
Try list ... to find functions!
Error in:
xl("x"); // label sumbu x ...
      ^
```

```
>yl("exp(x^3 - x)"); // label sumbu y
```

```
Function yl not found.
Try list ... to find functions!
Error in:
yl("exp(x^3 - x)"); // label sumbu y ...
      ^
```

```
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

ungsi eksponensial digambar dengan logaritma pada sumbu y untuk menunjukkan pertumbuhan cepat fungsi ini.

## Menggambar Fungsi Gamma

$$f(x) = G(x)$$

dengan Skala Logaritma pada Sumbu y

**Penjelasan :**

---

```
>reset; // menghapus semua plot sebelumnya
>plot2d("gamma(x)", 1, 10, logplot=1); // menggambar grafik fungsi gamma
```

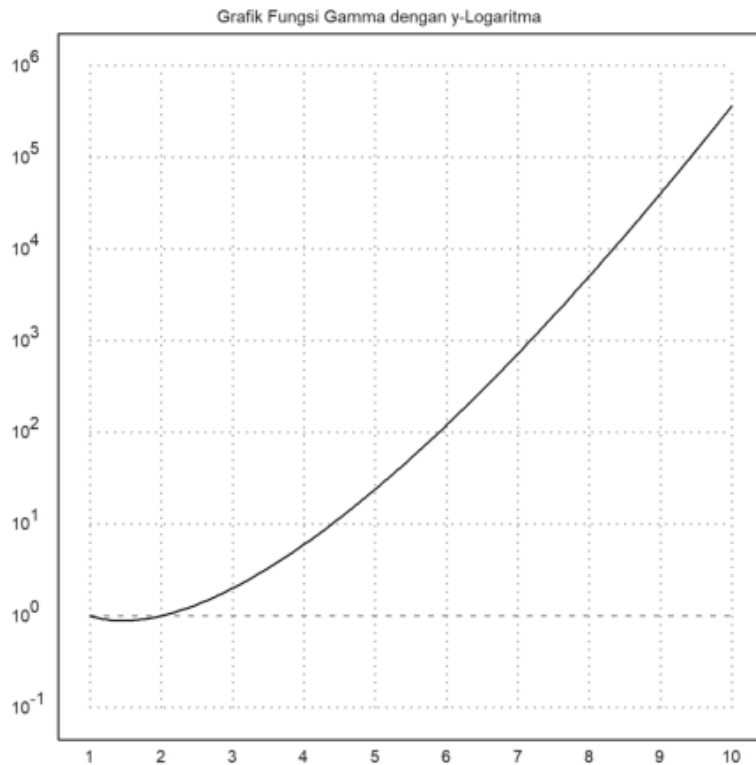
```
>title("Grafik Fungsi Gamma dengan y-Logaritma"); // judul
>xl("x"); // label sumbu x
```

```
Function xl not found.
Try list ... to find functions!
Error in:
xl("x"); // label sumbu x ...
      ^
```

```
>yl("Gamma(x)"); // label sumbu y
```

```
Function yl not found.
Try list ... to find functions!
Error in:
yl("Gamma(x)"); // label sumbu y ...
      ^
```

```
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

ungsi gamma adalah fungsi khusus yang biasanya menunjukkan nilai besar, sehingga penggunaan skala logaritma membantu dalam visualisasi.

---

## Rujukan Lengkap Fungsi plot2d() function plot2d (xv, yv, btest, a,

---

b, c, d, xmin, xmax, r, n, ..

logplot, kisi, bingkai, warna bingkai, kotak, warna, ketebalan, gaya, ..

otomatis, tambahkan, pengguna, delta, poin, titik tambahan, gaya titik, bilah, histogram, ..

distribusi, genap, langkah, sendiri, adaptif, rona, level, kontur, ..

nc, terisi, fillcolor, outline, title, xl, yl, maps, contourcolor, ..

contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral, ..

cgrid, vertikal, lebih kecil, dl, niveau, level)

Fungsi plot serbaguna untuk plot di pesawat (plot 2D). Fungsi ini dapat melakukan

plot fungsi satu variabel, plot data, kurva bidang, plot batang, kisi-kisi bilangan kompleks, dan plot implisit fungsi dua variabel.

Parameter

x,y : persamaan, fungsi atau vektor data

a,b,c,d : Area plot (default a=-2,b=2)

r : jika r diset, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r

r dapat berupa vektor [rx,ry] atau vektor

[rx1,rx2,ry1,ry2].

xmin,xmax : rentang parameter untuk kurva

auto : Menentukan y-range secara otomatis (default)

kuadrat : jika benar, coba pertahankan rentang x-y persegi

n : jumlah interval (default adaptif)

kisi : 0 = tidak ada kisi dan label,

1 = sumbu saja,

2 = grid normal (lihat di bawah untuk jumlah garis

grid)

3 = sumbu dalam

4 = tidak ada kisi-kisi

5 = kisi penuh termasuk margin

6 = kutu di bingkai

7 = sumbu saja

8 = sumbu saja, sub-centang

bingkai : 0 = tanpa bingkai

framecolor : warna bingkai dan kisi

margin : angka antara 0 dan 0,4 untuk margin di sekitar plot

warna : Warna kurva. Jika ini adalah vektor warna,

itu akan digunakan untuk setiap baris matriks plot. Dalam

kasus plot titik, itu harus berupa vektor kolom. Jika vektor baris atau matriks penuh warna digunakan untuk plot titik, itu akan digunakan untuk setiap titik data.

ketebalan: ketebalan garis untuk kurva

Nilai ini bisa lebih kecil dari 1 untuk garis yang sangat

tipis.

style : Plot style untuk garis, spidol, dan isian.

Untuk poin gunakan

"[]", "<>", ".", "..", "...",

"\*", "+", "|", "-", "o"

"[]#", "<>#", "o#" (bentuk terisi)

"[]w", "<>w", "ow" (tidak transparan)

Untuk penggunaan garis

"-", "--", "-.", ".", "-.-", "-.-", "->"

Untuk poligon terisi atau plot batang gunakan

"#", "#O", "O", "/", "\", "\/",

+", "|", "-", "t"

poin : plot titik tunggal alih-alih segmen garis  
 addpoints : jika benar, plot segmen garis dan titik  
 add : menambahkan plot ke plot yang ada  
 pengguna : aktifkan interaksi pengguna untuk fungsi  
 delta : ukuran langkah untuk interaksi pengguna  
 bar : plot batang (x adalah batas interval, y nilai interval)  
 histogram : memplot frekuensi x dalam n subinterval  
 distribusi=n : memplot distribusi x dengan n subinterval  
 even : gunakan nilai antar untuk histogram otomatis.  
 langkah : memplot fungsi sebagai fungsi langkah (langkah=1,2)  
 adaptif : gunakan plot adaptif (n adalah jumlah langkah minimal)  
 level : plot garis level dari fungsi implisit dua variabel  
 outline : menggambar batas rentang level.  
 Jika nilai level adalah matriks  $2 \times n$ , rentang level akan ditarik  
 dalam warna menggunakan gaya isian yang diberikan. Jika garis besar benar, itu  
 akan digambar dalam warna kontur. Dengan menggunakan fitur ini, wilayah  
 $f(x,y)$  antara batas dapat ditandai.  
 hue : tambahkan warna hue ke plot level untuk menunjukkan fungsinya

nilai

kontur : Gunakan plot level dengan level otomatis  
 nc : jumlah garis level otomatis  
 judul : judul plot (default "")  
 xl, yl : label untuk sumbu x dan y  
 lebih kecil : jika  $>0$ , akan ada lebih banyak ruang di sebelah kiri untuk label.  
 vertikal :

Mengaktifkan atau menonaktifkan label vertikal. Ini mengubah

variabel global

verticallabels secara lokal untuk satu plot. Nilai 1 hanya set

vertikal

teks, nilai 2 menggunakan label numerik vertikal pada sumbu y.

terisi : mengisi plot kurva  
 fillcolor : mengisi warna untuk bar dan kurva yang terisi  
 outline : batas poligon yang terisi  
 logplot : mengatur plot logaritma

1 = logplot di y,  
 2 = plot log di xy,  
 3 = logplot dalam x

memiliki :



Sebuah string, yang menunjuk ke rutinitas plot sendiri. Dengan >

pengguna, Anda mendapatkan

interaksi pengguna yang sama seperti di `plot2d`. Rentang akan diatur sebelum setiap panggilan ke fungsi Anda.

`peta` : ekspresi peta (0 lebih cepat), fungsi selalu dipetakan.

`contourcolor` : warna garis kontur

`contourwidth` : lebar garis kontur

`clipping` : mengaktifkan clipping (default adalah true)

`judul` :

Ini dapat digunakan untuk menggambarkan plot. Judul akan muncul di

atas

jalan cerita. Selain itu, label untuk sumbu x dan y dapat

ditambahkan dengan

`xl="string"` atau `yl="string"`. Label lain dapat ditambahkan dengan fungsi `label()` atau `labelbox()`. Judulnya bisa unicode string atau gambar rumus Lateks.

`jaringan` :

Menentukan jumlah garis grid untuk plot grid yang kompleks. Harus merupakan pembagi dari ukuran matriks dikurangi 1 (jumlah subinterval). `cgrid` dapat berupa vektor `[cx,cy]`.

Ringkasan

Fungsi dapat merencanakan

- ekspresi, koleksi panggilan atau fungsi dari satu variabel,
- kurva parametrik,
- data x terhadap data y,
- fungsi implisit,
- petak batang,
- jaringan kompleks,
- poligon.

Jika fungsi atau ekspresi untuk `xv` diberikan, `plot2d()` akan

menghitung nilai dalam rentang yang diberikan menggunakan fungsi atau ekspresi. Itu ekspresi harus berupa ekspresi dalam variabel x. Rentang harus didefinisikan dalam parameter a dan b kecuali rentang default harus digunakan. Rentang y akan dihitung secara otomatis, kecuali c dan d ditentukan, atau radius r, yang menghasilkan kisaran r,r untuk x dan y. Untuk plot fungsi, plot2d akan menggunakan evaluasi adaptif fungsi secara default. Untuk mempercepat plot untuk fungsi yang rumit, matikan ini dengan <adaptif, dan opsional mengurangi jumlah interval n. Selain itu, plot2d() akan secara default menggunakan pemetaan. Yaitu, itu akan menghitung elemen plot untuk elemen. Jika ekspresi atau fungsi Anda dapat menangani a vector x, Anda dapat menonaktifkannya dengan <maps untuk evaluasi yang lebih cepat.

Perhatikan bahwa plot adaptif selalu dihitung elemen untuk elemen.

Jika fungsi atau ekspresi untuk xv dan untuk yv ditentukan, plot2d() akan menghitung kurva dengan nilai xv sebagai koordinat x dan nilai yv sebagai koordinat y. Dalam hal ini, rentang harus didefinisikan untuk parameter menggunakan xmin, xmax. Ekspresi yang terkandung dalam string harus selalu ekspresi dalam variabel parameter x.

---

## Catatan Baru :

---

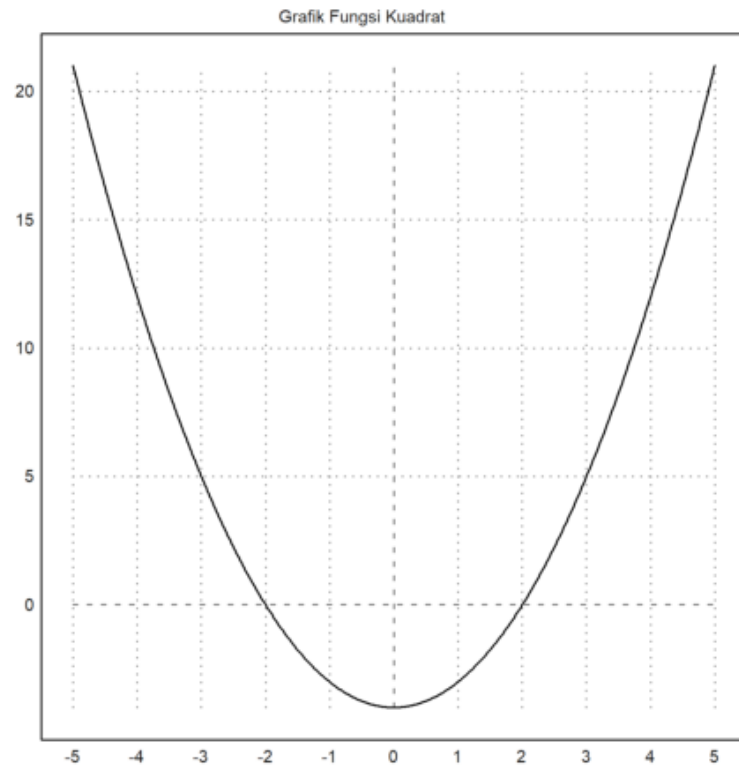
\* contoh penyelesaian masalah menggambar kurva/plot 2D \*\* 1 Menggambar Fungsi Kuadrat

$$f(x) = x^2 - 4$$

## Penyelesaian :

---

```
>reset; // menghapus semua plot sebelumnya
>plot2d("x^2 - 4", -5, 5); // menggambar grafik fungsi kuadrat
>title("Grafik Fungsi Kuadrat"); // memberikan judul pada plot
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

- reset: Menghapus semua pengaturan plot sebelumnya.
- plot2d("x^2 - 4", -5, 5): Menggambar grafik fungsi kuadrat dari -5 hingga 5
- Label dan judul ditambahkan untuk kejelasan.

---

2

---

Menggambar Fungsi Sinus

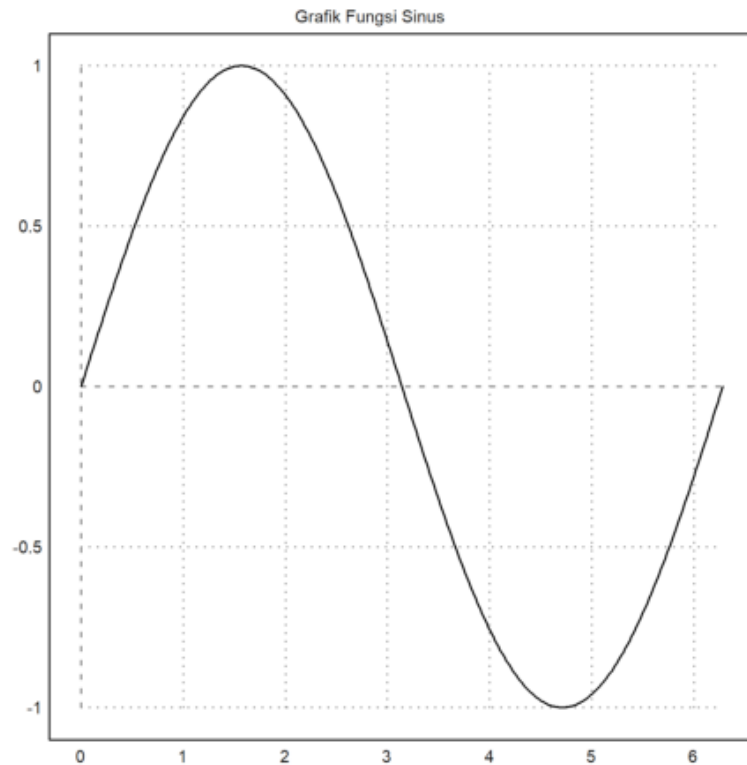
$$f(x) = \sin(x)$$

dengan Grid

**Penyelesaian :**

---

```
>reset; // menghapus semua plot sebelumnya
>plot2d("sin(x)", 0, 2*pi, grid=2); // menggambar grafik fungsi sinus dengan grid
>title("Grafik Fungsi Sinus"); // memberikan judul pada plot
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

- `plot2d("sin(x)", 0, 2*pi, grid=2)`: Menggambar grafik fungsi sinus dari 0 hingga  $2\pi$  dengan grid sumbu saja.
- label dan judul ditambahkan untuk menjelaskan grafik.

---

3

---

Menggambar Fungsi Eksponensial

$$f(x) = e^x$$

dengan Warna dan Ketebalan Garis

**Penyelesaian :**

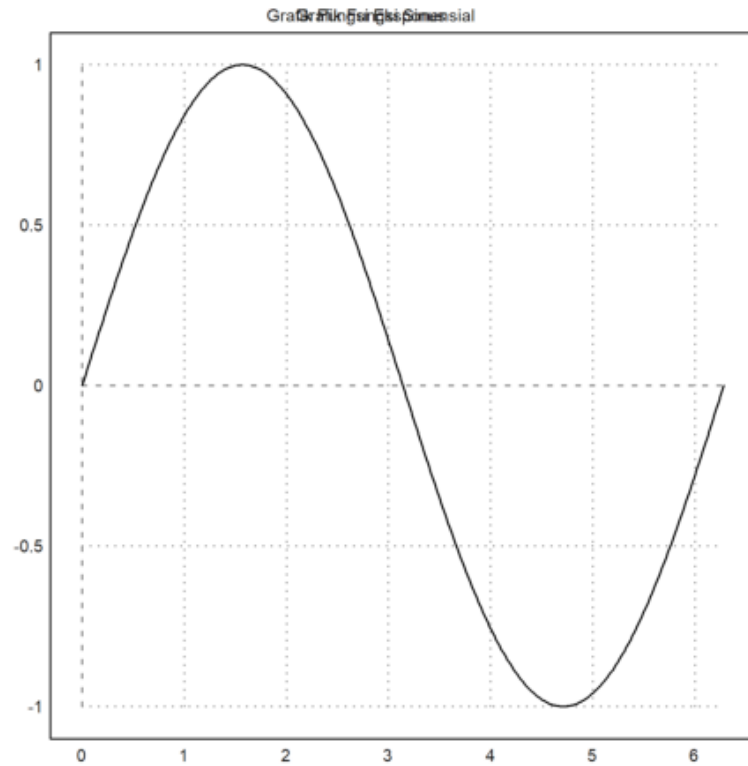
---

```
>reset; // menghapus semua plot sebelumnya
>plot2d("exp(x)", -2, 2, style="-", thickness=2, color="blue"); // menggambar grafik fungsi
```

```
Argument for color must be real!
Try "trace errors" to inspect local variables after errors.
plot2d:
  ccc=_color(color); l=_linewidth(thickness);
```

```
>title("Grafik Fungsi Eksponensial"); // memberikan judul pada plot
```

```
>insimg; // menampilkan gambar hasil plot
```



Penjelasan:

- `plot2d("exp(x)", -2, 2, style="-", thickness=2, color="blue")`: Menggambar grafik fungsi eksponensial dari -2 hingga 2 dengan garis biru dan ketebalan 2.
- Label dan judul ditambahkan untuk kejelasan.