

Tugas Latex Aplikasi Komputer



**Ardan Andhirta**  
22305141045  
Matematika E 2022

**PRODI MATEMATIKA**  
**DEPARTEMEN PENDIDIKAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS NEGERI YOGYAKARTA**  
**2022**

---

## DAFTAR ISI

1 KB Pekan 2: Pengenalan Software EMT	2
2 KB Pekan 3-4: Menggunakan EMT untuk menyelesaikan masalah-masalah Aljabar	17
3 KB Pekan 5-6: Menggunakan EMT untuk mengambar grafik 2 dimensi (2D)	83
4 KB Pekan 7-8: Menggunakan EMT untuk mengambar grafik 3 dimensi (3D)	161
5 KB Pekan 9-10: Menggunakan EMT untuk kalkulus	212
6 KB Pekan 11-12: Menggunakan EMT untuk Geometri	245
7 KB Pekan 13-14: Menggunakan EMT untuk Statistika	313

---

---

## BAB 1

---

# KB PEKAN 2: PENGENALAN SOFTWARE EMT

### Pendahuluan dan Pengenalan Cara Kerja EMT

---

Selamat datang! Ini adalah pengantar pertama ke Euler Math Toolbox (disingkat EMT atau Euler). EMT adalah sistem terintegrasi yang merupakan perpaduan kernel numerik Euler dan program komputer aljabar Maxima.

- Bagian numerik, GUI, dan komunikasi dengan Maxima telah dikembangkan oleh R. Grothmann, seorang profesor matematika di Universitas Eichstätt, Jerman. Banyak algoritma numerik dan pustaka software open source yang digunakan di dalamnya.

- Maxima adalah program open source yang matang dan sangat kaya untuk perhitungan simbolik dan aritmatika tak terbatas. Software ini dikelola oleh sekelompok pengembang di internet.

- Beberapa program lain (LaTeX, Povray, Tiny C Compiler, Python) dapat digunakan di Euler untuk memungkinkan perhitungan yang lebih cepat maupun tampilan atau grafik yang lebih baik.

Yang sedang Anda baca (jika dibaca di EMT) ini adalah berkas notebook di EMT. Notebook aslinya bawaan EMT (dalam bahasa Inggris) dapat dibuka melalui menu File, kemudian pilih "Open Tutorials and Examples", lalu pilih file "00 First Steps.en". Perhatikan, file notebook EMT memiliki ekstensi ".en". Melalui notebook ini Anda akan belajar menggunakan software Euler untuk menyelesaikan berbagai masalah matematika.

Panduan ini ditulis dengan Euler dalam bentuk notebook Euler, yang berisi teks (deskriptif), baris-baris perintah, tampilan hasil perintah (numerik, ekspresi matematika, atau gambar/plot), dan gambar yang disisipkan dari file gambar.

Untuk menambah jendela EMT, Anda dapat menekan [F11]. EMT akan menampilkan jendela grafik di layar desktop Anda. Tekan [F11] lagi untuk kembali ke tata letak favorit Anda. Tata letak disimpan untuk sesi berikutnya.

Anda juga dapat menggunakan [Ctrl]+[G] untuk menyembunyikan jendela grafik. Selanjutnya Anda dapat beralih antara grafik dan teks dengan tombol [TAB].

Seperti yang Anda baca, notebook ini berisi tulisan (teks) berwarna hijau, yang dapat Anda edit dengan meng-klik kanan teks atau tekan menu Edit -> Edit Comment atau tekan [F5], dan juga baris perintah EMT yang ditandai dengan ">" dan berwarna merah. Anda dapat menyisipkan baris perintah baru dengan cara menekan tiga tombol bersamaan: [Shift]+[Ctrl]+[Enter].

---

Komentar atau teks penjelasan dapat berisi beberapa "markup" dengan sintaks sebagai berikut.

- \* Judul - \*\* Sub-Judul - latex:  $F(x) = \int_a^x f(t) dt$  - mathjax:  $\frac{x^2 - 1}{x - 1} = x + 1$  - maxima :  $\text{integrate}(x^3, x) = \text{integrate}(x^3, x) + C$  - http : //www.euler-math-toolbox.de - See : http : //www.google.de | Google - image : hati.png -----

Hasil sintaks-sintaks di atas (tanpa diawali tanda strip) adalah sebagai berikut.

**Judul**

---

## Sub-Judul

---

$$F(x) = \int_a^x f(t) dt$$

$$\frac{x^2 - 1}{x - 1} = x + 1$$

$$\int x^3 dx = \frac{x^4}{4} + C$$

<http://www.euler-math-toolbox.de> See: <http://www.google.de> | Google



---

Gambar diambil dari folder images di tempat file notebook berada dan tidak dapat dibaca dari Web. Untuk "See:", tautan (URL)web lokal dapat digunakan.

Paragraf terdiri atas satu baris panjang di editor. Pergantian baris akan memulai baris baru. Paragraf harus dipisahkan dengan baris kosong.

```
>/ baris perintah diawali dengan >, komentar (keterangan) diawali dengan //
```

Mari kita tunjukkan cara menggunakan EMT sebagai kalkulator yang sangat canggih.

EMT berorientasi pada baris perintah. Anda dapat menuliskan satu atau lebih perintah dalam satu baris perintah. Setiap perintah harus diakhiri dengan koma atau titik koma.

- Titik koma menyembunyikan output (hasil) dari perintah. - sebuah koma mencetak hasilnya. - Setelah perintah terakhir, koma diasumsikan secara otomatis (boleh tidak ditulis).

Dalam contoh berikut, kita mendefinisikan variabel  $r$  yang diberi nilai 1,25. Output dari definisi ini adalah nilai variabel. Tetapi karena tanda titik koma, nilai ini tidak ditampilkan. Pada kedua perintah di belakangnya, hasil kedua perhitungan tersebut ditampilkan.

```
>r=1.25; pi*r^2, 2*pi*r
```

```
4.90873852123  
7.85398163397
```

### Latihan untuk Anda

---

- Sisipkan beberapa baris perintah baru - Tulis perintah-perintah baru untuk melakukan suatu perhitungan yang Anda inginkan, boleh menggunakan variabel, boleh tanpa variabel.

---

Beberapa catatan yang harus Anda perhatikan tentang penulisan sintaks perintah EMT.

- Pastikan untuk menggunakan titik desimal, bukan koma desimal untuk bilangan! - Gunakan \* untuk perkalian dan  $^n$  untuk eksponen (pangkat). - Seperti biasa, \* dan / bersifat lebih kuat daripada + atau -. -  ${}^m\sqrt{n}$  mengikuti lebih kuat dari \*, sehingga  $r^2$  merupakan rumus luas lingkaran. - Jika perlu, Anda harus menambahkan tanda kurung, seperti  $\pi(2^3)$ .

Perintah  $r = 1.25$  adalah menyimpan nilai ke variabel di EMT. Anda juga dapat menulis  $r := 1.25$  jika mau. Anda dapat menggunakan spasi sesuka Anda.

Anda juga dapat mengakhiri baris perintah dengan komentar yang diawali dengan dua garis miring (//).

```
>r := 1.25 // Komentar: Menggunakan := sebagai ganti =
```

```
1.25
```

Argumen atau input untuk fungsi ditulis di dalam tanda kurung.

```
>sin(45°), cos(pi), log(sqrt(E))
```

```
0.707106781187  
-1  
0.5
```

Seperti yang Anda lihat, fungsi trigonometri bekerja dengan radian, dan derajat dapat diubah dengan  $^{\circ}$ . Jika keyboard Anda tidak memiliki karakter derajat tekan [F7], atau gunakan fungsi deg() untuk mengonversi. EMT menyediakan banyak sekali fungsi dan operator matematika. Hampir semua fungsi matematika sudah tersedia di EMT. Anda dapat melihat daftar lengkap fungsi-fungsi matematika di EMT pada berkas Referensi (klik menu Help -> Reference)

Untuk membuat rangkaian komputasi lebih mudah, Anda dapat merujuk ke hasil sebelumnya dengan "perhitungan dalam baris perintah yang sama.

```
>(sqrt(5)+1)/2, %^2-%+1 // Memeriksa solusi x^2-x+1=0
```

1.61803398875

2

## Latihan untuk Anda

---

- Buka berkas Reference dan baca fungsi-fungsi matematika yang tersedia di EMT.
- Sisipkan beberapa baris perintah baru.
- Lakukan contoh-contoh perhitungan menggunakan fungsi-fungsi matematika di EMT.

## Satuan

---

EMT dapat mengubah unit satuan menjadi sistem standar internasional (SI). Tambahkan satuan di belakang angka untuk konversi sederhana.

```
>1miles // 1 mil = 1609,344 m
```

1609.344

Beberapa satuan yang sudah dikenal di dalam EMT adalah sebagai berikut. Semua unit diakhiri dengan tanda dolar (), namun boleh tidak perluditulis dengan mengaktifkan easyunits.

kilometer:= 1000; km:=kilometer; cm:=0.01; mm:= 0.001; minute:=60; min:= minute; minutes:= minute; hour:= 60\*minute; h:= hour; hours:= hour; day:= 24\*hour; days:= day; d:= day; year:= 365.2425\*day; years:= year; y:= year; inch:= 0.0254; in:=inch; feet:=12\*inch; foot:=feet; ft:=feet; yard:=3\*feet; yards:=yard; yd:=yard; mile:=1760\*yard; sec:= 1; ha:=10000; Ar:= 100; Tagwerk:=3408; Acre:= 4046.8564224; pt:=0.376mm;

Untuk konversi ke dan antar unit, EMT menggunakan operator khusus, yakni  $\rightarrow$ .

```
>4km -> miles, 4inch -> " mm"
```

2.48548476895

101.6 mm

## Format Tampilan Nilai

---

Akurasi internal untuk nilai bilangan di EMT adalah standar IEEE, sekitar 16 digit desimal. Aslinya, EMT tidak mencetak semua digit suatu bilangan. Ini untuk menghemat tempat dan agar terlihat lebih baik. Untuk mengatramilan satu bilangan, operator berikut dapat digunakan.

```
>pi
```

```
3.14159265359
```

```
>longest pi
```

```
3.141592653589793
```

```
>long pi
```

```
3.14159265359
```

```
>short pi
```

```
3.1416
```

```
>shortest pi
```

```
3.1
```

```
>fraction pi
```

```
312689/99532
```

```
>short 1200*1.03^10, long E, longest pi
```

```
1612.7
```

```
2.71828182846
```

```
3.141592653589793
```

Format aslinya untuk menampilkan nilai menggunakan sekitar 10 digit. Format tampilan nilai dapat diatur secara global atau hanya untuk satu nilai.

Anda dapat mengganti format tampilan bilangan untuk semua perintah selanjutnya. Untuk mengembalikan ke format aslinya dapat digunakan perintah "deformat" atau "reset".

```
>longestformat; pi, deformat; pi
```

```
3.141592653589793
```

```
3.14159265359
```

Kernel numerik EMT bekerja dengan bilangan titik mengambang (floating point) dalam presisi ganda IEEE (berbeda dengan bagian simbolik EMT). Hasil numerik dapat ditampilkan dalam bentuk pecahan.

```
>1/7+1/4, fraction %
```

```
0.392857142857  
11/28
```

## Perintah Multibaris

Perintah multi-baris membentang di beberapa baris yang terhubung dengan "..." di setiap akhir baris, kecuali baris terakhir. Untuk menghasilkan tanda pindah baris tersebut, gunakan tombol [Ctrl]+[Enter]. Ini akan menyambung perintah ke baris berikutnya dan menambahkan "..." di akhir baris sebelumnya. Untuk menggabungkan suatu baris ke baris sebelumnya, gunakan [Ctrl]+[Backspace].

Contoh perintah multi-baris berikut dapat dijalankan setiap kali kursor berada di salah satu barisnya. Ini juga menunjukkan bahwa ... harus berada di akhir suatu baris meskipun baris tersebut memuat komentar.

```
>a=4; b=15; c=2; // menyelesaikan a*x^2+b*x+c=0 secara manual ...
>D=sqrt(b^2/(a^2*4)-c/a); ...
>-b/(2*a) + D, ...
>-b/(2*a) - D
```

```
-0.138444501319
-3.61155549868
```

## Menampilkan Daftar Variabel

Untuk menampilkan semua variabel yang sudah pernah Anda definisikan sebelumnya (dan dapat dilihat kembali nilainya), gunakan perintah "listvar".

```
>listvar
```

r	1.25
a	4
b	15
c	2
D	1.73655549868123

Perintah listvar hanya menampilkan variabel buatan pengguna. Dimungkinkan untuk menampilkan variabel lain, dengan menambahkan string termuat di dalam nama variabel yang diinginkan.

Perlu Anda perhatikan, bahwa EMT membedakan huruf besar dan huruf kecil. Jadi variabel "d" berbeda dengan variabel "D".

Contoh berikut ini menampilkan semua unit yang diakhiri dengan "m" dengan mencari semua variabel yang berisi "m".

```
>listvar m$
```

km\$	1000
cm\$	0.01
mm\$	0.001
nm\$	1853.24496
gram\$	0.001
m\$	1
hquantum\$	6.62606957e-34
atm\$	101325

Untuk menghapus variabel tanpa harus memulai ulang EMT gunakan perintah "remvalue".

```
>remvalue a,b,c,D  
>D
```

Variable D not found!

Error in:

D ...  
^

## Menampilkan Panduan

---

Untuk mendapatkan panduan tentang penggunaan perintah atau fungsi di EMT, buka jendela panduan dengan menekan [F1] dan cari fungsinya. Anda juga dapat mengklik dua kali pada fungsi yang tertulis di baris perintah atau di teks untuk membuka jendela panduan.

Coba klik dua kali pada perintah "intrandom" berikut ini!

```
>intrandom(10,6)
```

[4, 2, 6, 2, 4, 2, 3, 2, 2, 6]

Di jendela panduan, Anda dapat mengklik kata apa saja untuk menemukan referensi atau fungsi.

Misalnya, coba klik kata "random" di jendela panduan. Kata tersebut boleh ada dalam teks atau di bagian "See:" pada panduan. Anda akan menemukan penjelasan fungsi "random", untuk menghasilkan bilangan acak berdistribusi uniform antara 0,0 dan 1,0. Dari panduan untuk "random" Anda dapat menampilkan panduan untuk fungsi "normal", dll.

```
>random(10)
```

[0.270906, 0.704419, 0.217693, 0.445363, 0.308411, 0.914541, 0.193585,  
0.463387, 0.095153, 0.595017]

```
>normal(10)
```

[-0.495418, 1.6463, -0.390056, -1.98151, 3.44132, 0.308178, -0.733427,  
-0.526167, 1.10018, 0.108453]

## Matriks dan Vektor

---

EMT merupakan suatu aplikasi matematika yang mengerti "bahasa matriks". Artinya, EMT menggunakan vektor dan matriks untuk perhitungan-perhitungan tingkat lanjut. Suatu vektor atau matriks dapat didefinisikan dengan tanda kurung siku. Elemen-elemennya dituliskan di dalam tanda kurung siku, antar elemen dalam satu baris dipisahkan oleh koma(), antar baris dipisahkan oleh titik koma();).

Vektor dan matriks dapat diberi nama seperti variabel biasa.

```
>v=[4,5,6,3,2,1]
```

[4, 5, 6, 3, 2, 1]

```
>A=[1,2,3;4,5,6;7,8,9]
```

1	2	3
4	5	6
7	8	9

Karena EMT mengerti bahasa matriks, EMT memiliki kemampuan yang sangat canggih untuk melakukan perhitungan matematis untuk masalah-masalah aljabar linier, statistika, dan optimisasi.

Vektor juga dapat didefinisikan dengan menggunakan rentang nilai dengan interval tertentu menggunakan tanda titik dua (:), seperti contoh berikut ini.

```
>c=1:5
```

```
[1, 2, 3, 4, 5]
```

```
>w=0:0.1:1
```

```
[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
```

```
>mean(w^2)
```

```
0.35
```

## Bilangan Kompleks

EMT juga dapat menggunakan bilangan kompleks. Tersedia banyak fungsi untuk bilangan kompleks di EMT. Bilangan imaginer

$$i = \sqrt{-1}$$

dituliskan dengan huruf I (huruf besar I), namun akan ditampilkan dengan huruf i (i kecil).

re(x) : bagian riil pada bilangan kompleks x. im(x) : bagian imaginer pada bilangan kompleks x. complex(x) : mengubah bilangan riil x menjadi bilangan kompleks. conj(x) : Konjugat untuk bilangan bilangan kompleks x. arg(x) : argumen (sudut dalam radian) bilangan kompleks x. real(x) : mengubah x menjadi bilangan riil.

Apabila bagian imaginer x terlalu besar, hasilnya akan menampilkan pesan kesalahan.

```
>sqrt(-1) // Error! >sqrt(complex(-1))
```

```
>z=2+3*I, re(z), im(z), conj(z), arg(z), deg(arg(z)), deg(arctan(3/2))
```

```
2+3i  
2  
3  
2-3i  
0.982793723247  
56.309932474  
56.309932474
```

```
>deg(arg(I)) // 90°
```

90

```
>sqrt(-1)
```

```
Floating point error!
Error in sqrt
Error in:
sqrt(-1) ...
^
```

```
>sqrt(complex(-1))
```

0+1i

EMT selalu menganggap semua hasil perhitungan berupa bilangan riil dan tidak akan secara otomatis mengubah ke bilangan kompleks.

Jadi akar kuadrat -1 akan menghasilkan kesalahan, tetapi akar kuadrat kompleks didefinisikan untuk bidang koordinat dengan cara seperti biasa. Untuk mengubah bilangan riil menjadi kompleks, Anda dapat menambahkan 0i atau menggunakan fungsi "complex".

```
>complex(-1), sqrt(%)
```

-1+0i  
0+1i

## Matematika Simbolik

EMT dapat melakukan perhitungan matematika simbolis (eksak) dengan bantuan software Maxima. Software Maxima otomatis sudah terpasang di komputer Anda ketika Anda memasang EMT. Meskipun demikian, Anda dapat juga memasang software Maxima tersendiri (yang terpisah dengan instalasi Maxima di EMT).

Pengguna Maxima yang sudah mahir harus memperhatikan bahwa terdapat sedikit perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks ekspresi simbolik di EMT.

Untuk melakukan perhitungan matematika simbolis di EMT, awali perintah Maxima dengan tanda "". Setiap ekspresi yang dimulai dengan "" adalah ekspresi simbolis dan dikerjakan oleh Maxima.

```
>& (a+b)^2
```

$$(b + a)^2$$

```
>&expand((a+b)^2), &factor(x^2+5*x+6)
```

$$\frac{2}{b^2 + 2ab + a^2}$$

$$(x^2 + 2)(x^2 + 3)$$

```
>&solve(a*x^2+b*x+c,x) // rumus abc
```

$$[x = \frac{(-\sqrt{b^2 - 4ac}) - b}{2a}, x = \frac{\sqrt{b^2 - 4ac} - b}{2a}]$$

```
>&(a^2-b^2)/(a+b), &ratsimp(%) // ratsimp menyederhanakan bentuk pecahan
```

$$\frac{a^2 - b^2}{b^2 + a^2}$$

$$a^2 - b^2$$

```
>10! // nilai faktorial (modus EMT)
```

3628800

```
>&10! //nilai faktorial (simbolik dengan Maxima)
```

3628800

Untuk menggunakan perintah Maxima secara langsung (seperti perintah pada layar Maxima) awali perintahnya dengan tanda ":" pada baris perintah EMT. Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "modus kompatibilitas").

```
>factor(1000) // mencari semua faktor 1000 (EMT)
```

[2, 2, 2, 5, 5, 5]

```
>::: factor(1000) // faktorisasi prima 1000 (dengan Maxima)
```

$$\begin{matrix} 3 & 3 \\ 2 & 5 \end{matrix}$$

```
>::: factor(20!)
```

$$\begin{matrix} 18 & 8 & 4 & 2 \\ 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 \end{matrix}$$

Jika Anda sudah mahir menggunakan Maxima, Anda dapat menggunakan sintaks asli perintah Maxima dengan menggunakan tanda "://" untuk mengawali setiap perintah Maxima di EMT. Perhatikan, harus ada spasi antara "://" dan perintahnya.

```
>::: binomial(5,2); // nilai C(5,2)
```

$$10$$

```
>::: binomial(m,4); // C(m,4)=m!/(4!(m-4)!)
```

$$\frac{(m - 3)(m - 2)(m - 1)m}{24}$$

```
>::: trigexpand(cos(x+y)); // rumus cos(x+y)=cos(x) cos(y)-sin(x)sin(y)
```

$$\cos(x) \cos(y) - \sin(x) \sin(y)$$

```
>::: trigexpand(sin(x+y));
```

$$\cos(x) \sin(y) + \sin(x) \cos(y)$$

```
>::: trigsimp(((1-sin(x)^2)*cos(x))/cos(x)^2+tan(x)*sec(x)^2) //menyederhanakan fungsi tri
```

$$\frac{\sin(x)^4 + \cos(x)^4}{\cos(x)^3}$$

Untuk menyimpan ekspresi simbolik ke dalam suatu variabel digunakan tanda "=".

```
>p1 &= (x^3+1) / (x+1)
```

$$\frac{x^3 + 1}{x + 1}$$

```
>&ratsimp(p1)
```

$$\frac{x^2 - x + 1}{x^2 + x + 1}$$

Untuk mensubstitusikan suatu nilai ke dalam variabel dapat digunakan perintah "with".

```
>&p1 with x=3 // (3^3+1) / (3+1)
```

7

```
>&p1 with x=a+b, &ratsimp(%) //substitusi dengan variabel baru
```

$$\frac{(b + a)^3 + 1}{b^3 + a^3 + 3ab^2 + 3a^2b + 1}$$

$$b^2 + (2a - 1)b^2 + a^2 - a + 1$$

```
>&diff(p1,x) //turunan p1 terhadap x
```

$$\begin{array}{r} 2 \quad \quad \quad 3 \\ 3x \quad \quad x + 1 \\ \hline x + 1 \quad \quad \quad (x + 1)^2 \end{array}$$

```
>&integrate(p1,x) // integral p1 terhadap x
```

$$\begin{array}{r} 3 \quad \quad \quad 2 \\ 2x - 3x + 6x \\ \hline 6 \end{array}$$

## Tampilan Matematika Simbolik dengan LaTeX

---

Anda dapat menampilkan hasil perhitungan simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukan hal ini, tambahkan tanda dolar () di depan dan di setiap intah Maxima. Perhatikan, hal ini hanya dapat menghasilkan

```
>$& (a+b) ^2
```

$$(b + a)^2$$

```
>$&expand( (a+b) ^2), $&factor(x^2+5*x+6)
```

$$(x + 2)(x + 3)$$

```
>$&solve(a*x^2+b*x+c,x) // rumus abc
```

$$\left[ x = \frac{-\sqrt{b^2 - 4ac} - b}{2a}, x = \frac{\sqrt{b^2 - 4ac} - b}{2a} \right]$$

```
>$& (a^2-b^2) / (a+b), $&ratsimp(%)
```

$$a - b$$

---

**Selamat Belajar dan Berlatih!**

Baik, itulah sekilas pengantar penggunaan software EMT. Masih banyak kemampuan EMT yang akan Anda pelajari dan praktikkan.

Sebagai latihan untuk memperlancar penggunaan perintah-perintah EMT yang sudah dijelaskan di atas, silakan Anda lakukan hal-hal sebagai berikut.

- Carilah soal-soal matematika dari buku-buku Matematika. - Tambahkan beberapa baris perintah EMT pada notebook ini. - Selesaikan soal-soal matematika tersebut dengan menggunakan EMT. Pilih soal-soal yang sesuai dengan perintah-perintah yang sudah dijelaskan dan dicontohkan di atas.

### Penyelesaian

```
>r :=7, pi* r ^ 2
```

```
7  
153.938040026
```

```
>sin(30°) + cos (45°)
```

```
1.20710678119
```

```
>log10(10)
```

```
1
```

```
>logbase (10, 5)
```

```
1.43067655807
```

```
>1±0.05
```

```
~0.95, 1.1~
```

```
>mean(normal(100000)) // 0-1-normal distributed values
```

```
0.000592833227266
```

```
>round(-1.9)
```

```
-2
```

$$8. \quad 88 : 4 \times 3 =$$

```
>88/4*3
```

66

6.  $\sin 120^\circ \cos 60^\circ$  sama dengan . . . .

```
>sin(120°) * cos(60°)
```

0.433012701892

---

---

## BAB 2

---

# KB PEKAN 3-4: MENGGUNAKAN EMT UNTUK MENYELESAIKAN MASALAH-MASALAH ALJABAR

---

---

### EMT untuk Perhitungan Aljabar

---

Pada notebook ini Anda belajar menggunakan EMT untuk melakukan berbagai perhitungan terkait dengan materi atau topik dalam Aljabar. Kegiatan yang harus Anda lakukan adalah sebagai berikut:

- Membaca secara cermat dan teliti notebook ini; - Menerjemahkan teks bahasa Inggris ke bahasa Indonesia; - Mencoba contoh-contoh perhitungan (perintah EMT) dengan cara meng ENTER setiap perintah EMT yang ada (pindahkan kursor ke baris perintah) - Jika perlu Anda dapat memodifikasi perintah yang ada dan memberikan keterangan/penjelasan tambahan terkait hasilnya. - Menyisipkan baris-baris perintah baru untuk mengerjakan soal-soal Aljabar dari file PDF yang saya berikan; - Memberi catatan hasilnya. - Jika perlu tuliskan soalnya pada teks notebook (menggunakan format LaTeX). - Gunakan tampilan hasil semua perhitungan yang eksak atau simbolik dengan format LaTeX. (Seperti contoh-contoh pada notebook ini.) **Contoh pertama**

---

Menyederhanakan bentuk aljabar:

$$6x^{-3}y^5 \times -7x^2y^{-9}$$

```
>$&6*x^(-3)*y^5*-7*x^2*y^(-9)
```

$$-\frac{42}{x y^4}$$

Menjabarkan:

$$(6x^{-3} + y^5)(-7x^2 - y^{-9})$$

```
> $&showev('expand((6*x^(-3)+y^5)*(-7*x^2-y^(-9))))
```

$$\text{expand}\left(\left(-\frac{1}{y^9} - 7x^2\right) \left(y^5 + \frac{6}{x^3}\right)\right) = -7x^2y^5 - \frac{1}{y^4} - \frac{6}{x^3y^9} - \frac{42}{x}$$

### Baris Perintah

---

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasil. Koma setelah perintah terakhir dapat dihilangkan.

Baris perintah berikut hanya akan mencetak hasil ekspresi, bukan tugas atau perintah format.

```
>r:=2; h:=4; pi*r^2*h/3
```

16.7551608191

Perintah harus dipisahkan dengan yang kosong. Baris perintah berikut mencetak dua hasilnya.

```
>pi*2*r*h, %+2*pi*r*h // Ingat tanda % menyatakan hasil perhitungan terakhir sebelumnya
```

50.2654824574  
100.530964915

Baris perintah dieksekusi dalam urutan yang ditekan pengguna kembali. Jadi Anda mendapatkan nilai baru setiap kali Anda menjalankan baris kedua.

```
>x := 1;  
>x := cos(x) // nilai cosinus (x dalam radian)
```

0.540302305868

```
>x := cos(x)
```

0.857553215846

Jika dua garis terhubung dengan "..." kedua garis akan selalu dieksekusi secara bersamaan.

```
>x := 1.5; ...  
>x := (x+2/x)/2, x := (x+2/x)/2, x := (x+2/x)/2,
```

1.41666666667  
1.41421568627  
1.41421356237

Ini juga merupakan cara yang baik untuk menyebarkan perintah panjang pada dua atau lebih baris. Anda dapat menekan Ctrl+Return untuk membagi garis menjadi dua pada posisi kursor saat ini, atau Ctrl+Back untuk menggabungkan garis.

Untuk melipat semua multi-garis tekan Ctrl + L. Kemudian garis-garis berikutnya hanya akan terlihat, jika salah satunya memiliki fokus. Untuk melipat satu multi-baris, mulailah baris pertama dengan "

```
>%+ x=4+5; ...
```

Garis yang dimulai dengan

81

Euler mendukung loop di baris perintah, selama mereka masuk ke dalam satu baris atau multi-baris. Dalam program, pembatasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut lihat pengantar berikut.

```
>x=1; for i=1 to 5; x := (x+2/x)/2, end; // menghitung akar 2
```

```
1.5  
1.4166666667  
1.41421568627  
1.41421356237  
1.41421356237
```

Tidak apa-apa untuk menggunakan multi-line. Pastikan baris diakhiri dengan "...".

```
>x := 1.5; // comments go here before the ...  
>repeat xnew:=(x+2/x)/2; until xnew~=:x; ...
```

Loop structure not complete, must be within one line!

```
> x := xnew; ...
```

Loop structure not complete, must be within one line!

```
>end; ...
```

"end" only allowed in functions or loops!

```
Error in :  
end; ...  
^
```

```
>x,
```

1.41421356237

Struktur bersyarat juga berfungsi.

```
>if E^pi>pi^E; then "Thought so!", endif;
```

Thought so!

Saat Anda menjalankan perintah, kursor dapat berada di posisi mana pun di baris perintah. Anda dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau Anda dapat mengklik ke bagian komentar di atas perintah untuk menuju ke perintah.

Saat Anda menggerakkan kursor di sepanjang garis, pasangan tanda kurung atau kurung buka dan tutup akan disorot. Juga, perhatikan baris status. Setelah kurung buka fungsi `sqrt()`, baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol kembali.

```
>sqrt(sin(10°)/cos(20°))
```

0.429875017772

Untuk melihat bantuan untuk perintah terbaru, buka jendela bantuan dengan F1. Di sana, Anda dapat memasukkan teks untuk dicari. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda dapat menekan escape untuk menghapus garis, atau untuk menutup jendela bantuan.

Anda dapat mengklik dua kali pada perintah apa pun untuk membuka bantuan untuk perintah ini. Coba klik dua kali perintah `exp` di bawah ini di baris perintah.

```
>exp(log(2.5))
```

2.5

Anda dapat menyalin dan menempel di Euler ke. Gunakan Ctrl-C dan Ctrl-V untuk ini. Untuk menandai teks, seret mouse atau gunakan shift bersama dengan tombol kursor apa pun. Selain itu, Anda dapat menyalin tanda kurung yang disorot.

## Sintaks Dasar

---

Euler tahu fungsi matematika biasa. Seperti yang Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilainya, atau gunakan fungsi `rad(x)`. Fungsi akar kuadrat disebut kuadrat dalam Euler. Tentu saja,  $x^{(1/2)}$  juga dimungkinkan. Untuk menyetel variabel, gunakan `"=` atau ":=. Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Tapi ruang antara perintah diharapkan.

Beberapa perintah dalam satu baris dipisahkan dengan `,` atau `";`. Titik koma menekan output dari perintah. Di akhir baris perintah `,` diasumsikan, jika `";` hilang.

```
>g:=9.81; t:=2.5; 1/2*g*t^2
```

30.65625

EMT menggunakan sintaks pemrograman untuk ekspresi. Memasuki

$$e^2 \cdot \left( \frac{1}{3 + 4 \log(0.6)} + \frac{1}{7} \right)$$

Anda harus mengatur tanda kurung yang benar dan menggunakan / untuk pecahan. Perhatikan tanda kurung yang disorot untuk bantuan. Perhatikan bahwa konstanta Euler e diberi nama E dalam EMT.

```
>E^2*(1/(3+4*log(0.6))+1/7)
```

8.77908249441

Untuk menghitung ekspresi rumit seperti

$$\left(\frac{\frac{1}{7} + \frac{1}{8} + 2}{\frac{1}{3} + \frac{1}{2}}\right)^2 \pi$$

Anda harus memasukkannya dalam bentuk baris.

```
>((1/7 + 1/8 + 2) / (1/3 + 1/2))^2 * pi
```

23.2671801626

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi bahwa braket penutup selesai. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil dari perhitungan ini adalah bilangan floating point. Secara default dicetak dengan akurasi sekitar 12 digit. Di baris perintah berikut, kita juga belajar bagaimana kita bisa merujuk ke hasil sebelumnya dalam baris yang sama.

```
>1/3+1/7, fraction %
```

0.47619047619

10/21

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terbuat dari operator dan fungsi. Jika perlu, itu harus berisi tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang braket adalah ide yang bagus. Perhatikan bahwa EMT menunjukkan tanda kurung buka dan tutup saat mengedit baris perintah.

```
>(cos(pi/4)+1)^3*(sin(pi/4)+1)^2
```

14.4978445072

Operator numerik Euler meliputi

+ unary atau operator plus - unary atau operator minus \*, / . produk matriks  $a^b$  *daya untuk positi faatau bilangan bulat b(a\*b) juga fungsi n! operator faktorial*

dan masih banyak lagi.

Berikut adalah beberapa fungsi yang mungkin Anda butuhkan. Ada banyak lagi.

sin, cos, tan, atan, asin, acos, rad, deg log, exp, log10, sqrt, logbase bin, logbin, logfac, mod, lantai, ceil, bulat, abs, tanda conj, re, im, arg beta, betai, gamma, complexgamma, ellrf, ellf, ellrd, elle bitand, bitor, bitxor, bitnot

Beberapa perintah memiliki alias, mis. Untuk log.

```
>ln(E^2), arctan(tan(0.5))
```

```
2  
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk menggunakan tanda kurung (kurung bulat), setiap kali ada keraguan tentang urutan eksekusi! Berikut ini tidak sama dengan  $(2^3)^4$ , yang merupakan *fault* untuk  $2^{34}$  di EMT (beberapa sistem numerik melakukan ini dengan benar).

```
>2^3^4, (2^3)^4, 2^(3^4)
```

```
2.41785e+024  
4096  
2.41785e+024
```

---

## Bilangan Asli

Tipe data utama dalam Euler adalah bilangan real. Real direpresentasikan dalam format IEEE dengan akurasi sekitar 16 digit desimal.

```
>longest 1/3
```

```
0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

```
>printdual(1/3)
```

```
1.01010101010101010101010101010101010101010101010101010101010101*2^-2
```

```
>printhex(1/3)
```

```
5.5555555555554*16^-1
```

---

## String

Sebuah string dalam Euler didefinisikan dengan "...".

```
>"A string can contain anything."
```

```
A string can contain anything.
```

String dapat digabungkan dengan `|` atau dengan `+`. Ini juga berfungsi dengan angka, yang dikonversi menjadi string dalam kasus itu.

```
>"The area of the circle with radius " + 2 + " cm is " + pi*4 + " cm^2."
```

The area of the circle with radius 2 cm is 12.5663706144 cm<sup>2</sup>.

Fungsi `print` juga mengonversi angka menjadi string. Ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan secara optimal satu unit.

```
>"Golden Ratio : " + print((1+sqrt(5))/2,5,0)
```

Golden Ratio : 1.61803

Ada string khusus tidak ada, yang tidak dicetak. Itu dikembalikan oleh beberapa fungsi, ketika hasilnya tidak masalah. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan pengembalian.)

```
>none
```

Untuk mengonversi string menjadi angka, cukup evaluasi saja. Ini juga berfungsi untuk ekspresi (lihat di bawah).

```
>"1234.5"()
```

1234.5

Untuk mendefinisikan vektor string, gunakan notasi vektor [...].

```
>v := ["affe", "charlie", "bravo"]
```

```
affe  
charlie  
bravo
```

Vektor string kosong dilambangkan dengan `[none]`. Vektor string dapat digabungkan.

```
>w := [none]; w | v | v
```

```
affe  
charlie  
bravo  
affe  
charlie  
bravo
```

String dapat berisi karakter Unicode. Secara internal, string ini berisi kode UTF-8. Untuk menghasilkan string seperti itu, gunakan `u"..."` dan salah satu entitas HTML.

String Unicode dapat digabungkan seperti string lainnya.

```
>u"&alpha; = " + 45 + u"&deg;" // pdfLaTeX mungkin gagal menampilkan secara benar
```

Variable u not found!

Error in :

```
u"&alpha; = " + 45 + u"&deg;" // pdfLaTeX mungkin gagal menampilkan secara benar  
^
```

I

Dalam komentar, entitas yang sama seperti alpha;, beta;, dll dapat digunakan. Ini mungkin alternatif cepat untuk Lateks. (Lebih detail di komentar di bawah).

Ada beberapa fungsi untuk membuat atau menganalisis string unicode. Fungsi strtochar() akan mengenali string Unicode, dan menerjemahkannya dengan benar.

```
>v=strtochar(u"&Auml; is a German letter")
```

Closing bracket missing in function call!

Error in :

```
v=strtochar(u"&Auml; is a German letter")  
^
```

Hasilnya adalah vektor angka Unicode. Fungsi kebalikannya adalah chartoutf().

```
>v[1]=strtochar(u"&Uuml;")[1]; chartoutf(v)
```

Closing bracket missing in function call!

Error in :

```
v[1]=strtochar(u"&Uuml;")[1]; chartoutf(v)  
^
```

Fungsi utf() dapat menerjemahkan string dengan entitas dalam variabel menjadi string Unicode.

```
>s="We have &alpha;=&beta; ."; utf(s) // pdfLaTeX mungkin gagal menampilkan secara benar
```

Function utf not found.

Try list ... to find functions!

Error in :

```
s="We have &alpha;=&beta; ."; utf(s) // pdfLaTeX mungkin gagal menampilkan secara benar  
^
```

Dimungkinkan juga untuk menggunakan entitas numerik.

```
>u"\u00f6;hnliches"
```

```
Variable u not found!
```

```
Error in :  
u" ;hnliches"  
^
```

## Nilai Boolean

---

Nilai Boolean direpresentasikan dengan 1=true atau 0=false dalam Euler. String dapat dibandingkan, seperti halnya angka.

```
>2<1, "apel"<"banana"
```

```
0  
1
```

"dan" adalah operator "" dan "atau" adalah operator "||", seperti dalam bahasa C. (Kata-kata "dan" dan "atau" hanya dapat digunakan dalam kondisi untuk "jika".)

```
>2<E && E<3
```

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

```
>(1:10)>5, nonzeros(%)
```

```
[0, 0, 0, 0, 0, 1, 1, 1, 1, 1]  
[6, 7, 8, 9, 10]
```

Anda dapat menggunakan fungsi bukan nol() untuk mengekstrak elemen tertentu dari vektor. Dalam contoh, kami menggunakan isprime bersyarat(n).

```
>N=2|3:2:99 // N berisi elemen 2 dan bilangan2 ganjil dari 3 s.d. 99
```

```
[2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29,  
31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57,  
59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85,  
87, 89, 91, 93, 95, 97, 99]
```

```
>N[nonzeros(isprime(N))] //pilih anggota2 N yang prima
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,  
53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

## Format Keluaran

---

Format output default EMT mencetak 12 digit. Untuk memastikan bahwa kami melihat default, kami mengatur ulang format.

```
>defformat; pi
```

```
3.14159265359
```

Secara internal, EMT menggunakan standar IEEE untuk bilangan ganda dengan sekitar 16 digit desimal. Untuk melihat jumlah digit penuh, gunakan perintah "format terpanjang", atau kita gunakan operator "terpanjang" untuk menampilkan hasil dalam format terpanjang.

```
>longest pi
```

```
3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

```
3.243F6A8885A30*16^0
```

Format output dapat diubah secara permanen dengan perintah format.

```
>format(12,5); 1/3, pi, sin(1)
```

```
0.33333  
3.14159  
0.84147
```

Standarnya adalah format (12).

```
>format(12); 1/3
```

```
0.333333333333
```

Fungsi seperti "shortestformat", "shortformat", "longformat" bekerja untuk vektor dengan cara berikut.

```
>shortestformat; random(3,8)
```

```
0.6554 0.201 0.8936 0.2819 0.525 0.3141 0.4446 0.2995  
0.2827 0.8832 0.2709 0.7044 0.2177 0.4454 0.3084 0.9145  
0.1936 0.4634 0.09515 0.595 0.4312 0.7287 0.4652 0.323
```

Format default untuk skalar adalah format (12). Tapi ini bisa diubah.

```
>setscalarformat(5); pi
```

```
3.1416
```

Fungsi "format terpanjang" mengatur format skalar juga.

```
>longestformat; pi
```

```
3.141592653589793
```

Untuk referensi, berikut adalah daftar format output yang paling penting.

format terpendek format pendek format panjang, format terpanjang format(panjang,digit) format baik(panjang) fracformat (panjang) mengubah bentuk

Akurasi internal EMT adalah sekitar 16 tempat desimal, yang merupakan standar IEEE. Angka disimpan dalam format internal ini.

Tetapi format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

```
>format(10,5); pi
```

```
3.14159
```

Standarnya adalah deformat().

```
>deformat; // default
```

Ada operator pendek yang hanya mencetak satu nilai. Operator "terpanjang" akan mencetak semua digit angka yang valid.

```
>longest pi^2/2
```

```
4.934802200544679
```

Ada juga operator pendek untuk mencetak hasil dalam format pecahan. Kami sudah menggunakan di atas.

```
>fraction 1+1/2+1/3+1/4
```

```
25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0,1 tidak akan direpresentasikan dengan tepat. Kesalahan bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

```
>longest 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
```

```
-1.110223024625157e-016
```

Tetapi dengan "format panjang" default Anda tidak akan melihat ini. Untuk kenyamanan, output dari angka yang sangat kecil adalah 0.

```
>0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
```

0

## Ekspresi

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda bermaksud menggunakan string sebagai ekspresi, gunakan konvensi untuk menamakannya "fx" atau "fxy" dll. Ekspresi lebih diutamakan daripada fungsi. Variabel global dapat digunakan dalam evaluasi.

```
>r:=2; fx:="pi*r^2"; longest fx()
```

12.56637061435917

Parameter ditetapkan ke x, y, dan z dalam urutan itu. Parameter tambahan dapat ditambahkan menggunakan parameter yang ditetapkan.

```
>fx:="a*sin(x)^2"; fx(5,a=-1)
```

-0.919535764538

Perhatikan bahwa ekspresi akan selalu menggunakan variabel global, bahkan jika ada variabel dalam fungsi dengan nama yang sama. (Jika tidak, evaluasi ekspresi dalam fungsi dapat memberikan hasil yang sangat membingungkan bagi pengguna yang memanggil fungsi tersebut.)

```
>at:=4; function f(expr,x,at) := expr(x); ...
>f("at*x^2",3,5) // computes 4*3^2 not 5*3^2
```

36

Jika Anda ingin menggunakan nilai lain untuk "at" daripada nilai global, Anda perlu menambahkan "at=value".

```
>at:=4; function f(expr,x,a) := expr(x,at=a); ...
>f("at*x^2",3,5)
```

45

Untuk referensi, kami berkomentar bahwa koleksi panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi kita bisa membuat contoh di atas sebagai berikut.

```
>at:=4; function f(expr,x) := expr(x); ...
>f({ {"at*x^2",at=5} },3)
```

```
Syntax error in expression, or unfinished expression!
```

```
Error in :  
f({{"at*x^2",at=5}},3)  
^
```

Ekspresi dalam x sering digunakan seperti fungsi. Perhatikan bahwa mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global menghapus variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

```
>f &= 5*x;  
>function f(x) := 6*x;  
>f(2)
```

12

Dengan cara konvensi, ekspresi simbolik atau numerik harus diberi nama fx, fxy dll. Skema penamaan ini tidak boleh digunakan untuk fungsi.

```
>fx &= diff(x^x,x); $&fx
```

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

Bentuk khusus dari ekspresi memungkinkan variabel apa pun sebagai parameter tanpa nama untuk evaluasi ekspresi, bukan hanya "x", "y" dll. Untuk ini, mulai ekspresi dengan "@(variabel) ...".

```
>"@(a,b) a^2+b^2", %(4,5)
```

```
@(a,b) a^2+b^2  
41
```

Ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam "x".

Cara paling dasar untuk mendefinisikan fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolis atau numerik. Jika variabel utama adalah x, ekspresi dapat dievaluasi seperti fungsi.

Seperti yang Anda lihat dalam contoh berikut, variabel global terlihat selama evaluasi.

```
>fx &= x^3-a*x; ...  
>a=1.2; fx(0.5)
```

-0.475

Semua variabel lain dalam ekspresi dapat ditentukan dalam evaluasi menggunakan parameter yang ditetapkan.

```
>fx(0.5,a=1.1)
```

-0.425

Sebuah ekspresi tidak perlu simbolis. Ini diperlukan, jika ekspresi berisi fungsi, yang hanya diketahui di kernel numerik, bukan di Maxima.

## Matematika Simbolik

EMT melakukan matematika simbolis dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut, atau telusuri referensi untuk Maxima. Para ahli di Maxima harus mencatat bahwa ada perbedaan sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik di EMT.

Matematika simbolik terintegrasi dengan mulus ke dalam Euler dengan . Ekspresi apa pun yang dimulai dengan adalah ekspresi simbolis. Itu dievaluasi dan dicetak oleh Maxima.

Pertama-tama, Maxima memiliki aritmatika "tak terbatas" yang dapat menangani angka yang sangat besar.

```
> $& 44 !
```

```
26582715747884487680436258110146158903196385280000000000
```

Dengan cara ini, Anda dapat menghitung hasil yang besar dengan tepat. Mari kita hitung lateks:  $C(44,10) = \frac{44!}{34! \cdot 10!}$

```
> $& 44! / (34!*10!) // nilai C(44,10)
```

```
2481256778
```

Tentu saja, Maxima memiliki fungsi yang lebih efisien untuk ini (seperti halnya bagian numerik dari EMT).

```
> $binomial(44,10) // menghitung C(44,10) menggunakan fungsi binomial()
```

```
2481256778
```

Untuk mempelajari lebih lanjut tentang fungsi tertentu klik dua kali di atasnya. Misalnya, coba klik dua kali pada "binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima seperti yang disediakan oleh penulis program itu.

Anda akan belajar bahwa yang berikut ini juga berfungsi.

$$C(x, 3) = \frac{x!}{(x-3)!3!} = \frac{(x-2)(x-1)x}{6}$$

```
> $binomial(x, 3) // C(x, 3)
```

$$\frac{(x-2)(x-1)x}{6}$$

Jika Anda ingin mengganti x dengan nilai tertentu, gunakan "dengan".

```
>${binomial(x, 3) with x=10 // substitusi x=10 ke C(x, 3)}
```

120

Dengan begitu Anda dapat menggunakan solusi persamaan dalam persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasan untuk ini adalah bendera simbolis khusus dalam string.

Seperti yang akan Anda lihat pada contoh sebelumnya dan berikut, jika Anda telah menginstal LaTeX, Anda dapat mencetak ekspresi simbolis dengan Lateks. Jika tidak, perintah berikut akan mengeluarkan pesan kesalahan.

Untuk mencetak ekspresi simbolis dengan LaTeX, gunakan *didepan*(atau *Anda dapat menghilangkan*) sebelum perintah. Jangan jika Anda tidak menginstal LaTeX.

```
>$(3+x)/(x^2+1)
```

$$\frac{x + 3}{x^2 + 1}$$

Ekspresi simbolik diuraikan oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat menyertakan ekspresi dalam "...". Untuk menggunakan lebih dari ekspresi sederhana adalah mungkin, tetapi sangat tidak disarankan.

```
>&"v := 5; v^2"
```

25

Untuk kelengkapan, kami menyatakan bahwa ekspresi simbolik dapat digunakan dalam program, tetapi perlu diapit dalam tanda kutip. Selain itu, jauh lebih efektif untuk memanggil Maxima pada waktu kompilasi jika memungkinkan.

```
>${expand((1+x)^4), ${factor(diff(% ,x))} // diff: turunan, factor: faktor}
```

$$x^4 + 4x^3 + 6x^2 + 4x + 1$$

$$4(x + 1)^3$$

Sekali lagi,

Untuk mempermudah, kami menyimpan solusi ke variabel simbolik. Variabel simbolik didefinisikan dengan "=".

```
>fx &= (x+1)/(x^4+1); $&fx
```

$$\frac{x+1}{x^4+1}$$

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

```
> $&factor (diff (fx, x))
```

$$\frac{-3x^4 - 4x^3 + 1}{(x^4 + 1)^2}$$

Masukan langsung dari perintah Maxima juga tersedia. Mulai baris perintah dengan "::". Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "mode kompatibilitas").

```
>&factor (20!)
```

2432902008176640000

```
>::: factor (10!)
```

$$\begin{matrix} 8 & 4 & 2 \\ 2 & 3 & 5 & 7 \end{matrix}$$

```
>::: factor (20!)
```

$$\begin{matrix} 18 & 8 & 4 & 2 \\ 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 \end{matrix}$$

Jika Anda ahli dalam Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukannya dengan ":::".

```
>::: av:g$ av^2;
```

$$g^2$$

```
>fx &= x^3*exp (x), $fx
```

$$\begin{matrix} 3 & x \\ x & E \end{matrix}$$

$$x^3 e^x$$

Variabel tersebut dapat digunakan dalam ekspresi simbolik lainnya. Perhatikan, bahwa dalam perintah berikut sisi kanan = dievaluasi sebelum penugasan ke Fx.

```
>&(fx with x=5), $%, &float(%)
```

$$\begin{array}{r} 5 \\ 125 \text{ E} \end{array}$$

$$125 e^5$$

$$18551.64488782208$$

```
>fx(5)
```

$$18551.6448878$$

Untuk evaluasi ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator "with". Baris perintah berikut juga menunjukkan bahwa Maxima dapat mengevaluasi ekspresi secara numerik dengan float().

```
>&(fx with x=10)-(fx with x=5), &float(%)
```

$$\begin{array}{r} 10 \\ 1000 \text{ E} \end{array} - \begin{array}{r} 5 \\ 125 \text{ E} \end{array}$$

$$2.2007914149918895\text{E}+7$$

```
>$factor(diff(fx,x,2))
```

$$x (x^2 + 6x + 6) e^x$$

Untuk mendapatkan kode Lateks untuk ekspresi, Anda dapat menggunakan perintah tex.

```
>tex(fx)
```

$$x^3 \backslash, e^{\{x\}}$$

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

0.206090158838

Dalam ekspresi simbolis, ini tidak berfungsi, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks "with" (bentuk yang lebih bagus dari perintah at(...) dari Maxima).

```
>$&fx with x=1/2
```

$$\frac{\sqrt{e}}{8}$$

Penugasan juga bisa bersifat simbolis.

```
>$&fx with x=1+t
```

$$(t + 1)^3 e^{t+1}$$

Perintah solve memecahkan ekspresi simbolik untuk variabel di Maxima. Hasilnya adalah vektor solusi.

```
>$&solve(x^2+x=4,x)
```

$$\left[ x = \frac{-\sqrt{17} - 1}{2}, x = \frac{\sqrt{17} - 1}{2} \right]$$

Bandingkan dengan perintah numerik "selesaikan" di Euler, yang membutuhkan nilai awal, dan secara opsional nilai target.

```
>solve("x^2+x",1,y=4)
```

1.56155281281

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolis. Euler akan membaca tugas x= dll. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat memberikan Maxima menemukan nilai numerik.

```
>sol &= solve(x^2+2*x=4,x); $&sol, sol(), $&float(sol)
```

$$\left[ x = -\sqrt{5} - 1, x = \sqrt{5} - 1 \right]$$

[-3.23607, 1.23607]

$$[x = -3.23606797749979, x = 1.23606797749979]$$

Untuk mendapatkan solusi simbolis tertentu, seseorang dapat menggunakan "with" dan index.

```
> $& solve(x^2+x=1, x), x2 &= x with %[2]; $&x2
```

$$\left[ x = \frac{-\sqrt{5} - 1}{2}, x = \frac{\sqrt{5} - 1}{2} \right]$$
$$\frac{\sqrt{5} - 1}{2}$$

Untuk menyelesaikan sistem persamaan, gunakan vektor persamaan. Hasilnya adalah vektor solusi.

```
> sol &= solve([x+y=3, x^2+y^2=5], [x, y]); $&sol, $&x*y with sol[1]
```

$$[[x = 2, y = 1], [x = 1, y = 2]]$$

2

Ekspresi simbolis dapat memiliki bendera, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, yang lain tidak. Bendera ditambahkan dengan "|" (bentuk yang lebih bagus dari "ev(...,flags)")

```
> $& diff((x^3-1)/(x+1), x) //turunan bentuk pecahan
```

$$\frac{3x^2}{x+1} - \frac{x^3 - 1}{(x+1)^2}$$

```
> $& diff((x^3-1)/(x+1), x) | ratsimp //menyederhanakan pecahan
```

$$\frac{2x^3 + 3x^2 + 1}{x^2 + 2x + 1}$$

```
> $& factor(%)
```

$$\frac{2x^3 + 3x^2 + 1}{(x+1)^2}$$

## Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "fungsi". Ini bisa berupa fungsi satu baris atau fungsi multibaris. Fungsi satu baris dapat berupa numerik atau simbolis. Fungsi satu baris numerik didefinisikan oleh ":=".

```
>function f(x) := x*sqrt(x^2+1)
```

Untuk gambaran umum, kami menunjukkan semua kemungkinan definisi untuk fungsi satu baris. Suatu fungsi dapat dievaluasi sama seperti fungsi Euler bawaan lainnya.

```
>f(2)
```

4.472135955

Fungsi ini akan bekerja untuk vektor juga, dengan mematuhi bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi divektorkan.

```
>f(0:0.1:1)
```

```
[0, 0.100499, 0.203961, 0.313209, 0.430813, 0.559017, 0.699714,  
0.854459, 1.0245, 1.21083, 1.41421]
```

Fungsi dapat diplot. Alih-alih ekspresi, kita hanya perlu memberikan nama fungsi.

Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus diberikan dalam string.

```
>solve("f",1,y=1)
```

0.786151377757

Secara default, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "menimpa". Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah untuk fungsi lain tergantung pada fungsi tersebut.

Anda masih dapat memanggil fungsi bawaan sebagai "...", jika itu adalah fungsi di *inti Euler*.

```
>function overwrite sin (x) := _sin(x°) // redine sine in degrees  
>sin(45)
```

0.707106781187

Lebih baik kita menghapus redefinisi dosa ini.

```
>forget sin; sin(pi/4)
```

0.707106781187

## Parameter Default

---

Fungsi numerik dapat memiliki parameter default.

```
>function f(x,a=1) := a*x^2
```

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menyetelnya akan menimpa nilai default.

```
>f(4,5)
```

80

Parameter yang ditetapkan menimpanya juga. Ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

```
>f(4,a=1)
```

16

Jika suatu variabel bukan parameter, itu harus global. Fungsi satu baris dapat melihat variabel global.

```
>function f(x) := a*x^2  
>a=6; f(2)
```

24

Tetapi parameter yang ditetapkan menimpa nilai global.

Jika argumen tidak ada dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut harus dideklarasikan dengan ":=!"

```
>f(2,a:=5)
```

20

Fungsi simbolis didefinisikan dengan "=". Mereka didefinisikan dalam Euler dan Maxima, dan bekerja di kedua dunia. Ekspresi yang mendefinisikan dijalankan melalui Maxima sebelum definisi.

```
>function g(x) &= x^3-x*exp(-x); $&g(x)
```

$$x^3 - x e^{-x}$$

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

```
> $&diff(g(x), x), $&% with x=4/3
```

$$x e^{-x} - e^{-x} + 3 x^2$$

$$\frac{e^{-\frac{4}{3}}}{3} + \frac{16}{3}$$

Mereka juga dapat digunakan dalam ekspresi numerik. Tentu saja, ini hanya akan berfungsi jika EMT dapat menginterpretasikan semua yang ada di dalam fungsi tersebut.

```
> g(5+g(1))
```

$$178.635099908$$

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

```
> function G(x) &= factor(integrate(g(x), x)); $&G(c) // integrate: mengintegralkan
```

$$\frac{e^{-c} (c^4 e^c + 4 c + 4)}{4}$$

```
> solve(&g(x), 0.5)
```

$$0.703467422498$$

Berikut ini juga berfungsi, karena Euler menggunakan ekspresi simbolis dalam fungsi  $g$ , jika tidak menemukan variabel simbolik  $g$ , dan jika ada fungsi simbolis  $g$ .

```
> solve(&g, 0.5)
```

$$0.703467422498$$

```
> function P(x, n) &= (2*x-1)^n; $&P(x, n)
```

$$(2x - 1)^n$$

```
> function Q(x, n) &= (x+2)^n; $&Q(x, n)
```

$$(x + 2)^n$$

```
>${&P(x, 4)}, ${&expand(%)}
```

$$(2x - 1)^4$$

$$16x^4 - 32x^3 + 24x^2 - 8x + 1$$

```
>P(3, 4)
```

625

```
>${&P(x, 4) + Q(x, 3)}, ${&expand(%)}
```

$$(2x - 1)^4 + (x + 2)^3$$

$$16x^4 - 31x^3 + 30x^2 + 4x + 9$$

```
>${&P(x, 4) - Q(x, 3)}, ${&expand(%)}, ${&factor(%)}
```

$$(2x - 1)^4 - (x + 2)^3$$

$$16x^4 - 33x^3 + 18x^2 - 20x - 7$$

$$16x^4 - 33x^3 + 18x^2 - 20x - 7$$

```
>${&P(x, 4) * Q(x, 3)}, ${&expand(%)}, ${&factor(%)}
```

$$(x + 2)^3 (2x - 1)^4$$

$$16x^7 + 64x^6 + 24x^5 - 120x^4 - 15x^3 + 102x^2 - 52x + 8$$

$$(x + 2)^3 (2x - 1)^4$$

```
>${&P(x, 4) / Q(x, 1)}, ${&expand(%)}, ${&factor(%)}
```

$$\frac{(2x - 1)^4}{x + 2}$$

$$\frac{16x^4}{x + 2} - \frac{32x^3}{x + 2} + \frac{24x^2}{x + 2} - \frac{8x}{x + 2} + \frac{1}{x + 2}$$

$$\frac{(2x - 1)^4}{x + 2}$$

```
>function f(x) &= x^3-x; $&f(x)
```

$$x^3 - x$$

Dengan = fungsinya simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&integrate(f(x),x)
```

$$\frac{x^4}{4} - \frac{x^2}{2}$$

Dengan := fungsinya numerik. Contoh yang baik adalah integral tak tentu seperti

$$f(x) = \int_1^x t^t dt,$$

yang tidak dapat dinilai secara simbolis.

Jika kita mendefinisikan kembali fungsi dengan kata kunci "peta" dapat digunakan untuk vektor x. Secara internal, fungsi dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam vektor.

```
>function map f(x) := integrate("x^x",1,x)
>f(0:0.5:2)
```

```
[-0.783431, -0.410816, 0, 0.676863, 2.05045]
```

Fungsi dapat memiliki nilai default untuk parameter.

```
>function mylog (x,base=10) := ln(x)/ln(base);
```

Sekarang fungsi dapat dipanggil dengan atau tanpa parameter "basis".

```
>mylog(100), mylog(2^6.7,2)
```

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

```
>mylog(E^2,base=E)
```

```
2
```

Seringkali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individual di tempat lain. Ini dimungkinkan dengan parameter vektor.

```
>function f([a,b]) &= a^2+b^2-a*b+b; $&f(a,b), $&f(x,y)
```

$$b^2 - a b + b + a^2$$

$$y^2 - x y + y + x^2$$

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.

Tetapi fungsinya juga dapat digunakan untuk vektor numerik.

```
>v=[3,4]; f(v)
```

17

Ada juga fungsi simbolis murni, yang tidak dapat digunakan secara numerik.

```
>function lapl(expr,x,y) &&= diff(expr,x,2)+diff(expr,y,2)//turunan parsial kedua
```

$$\text{diff(expr, } y, \ 2) \ + \ \text{diff(expr, } x, \ 2)$$

```
>$&realpart((x+I*y)^4), $&lapl(% ,x,y)
```

$$y^4 - 6 x^2 y^2 + x^4$$

0

Tetapi tentu saja, mereka dapat digunakan dalam ekspresi simbolik atau dalam definisi fungsi simbolik.

```
>function f(x,y) &= factor(lapl((x+y^2)^5,x,y)); $&f(x,y)
```

$$10 \left(y^2 + x\right)^3 \left(9 y^2 + x + 2\right)$$

Untuk meringkas

- = mendefinisikan fungsi simbolis, - := mendefinisikan fungsi numerik, - = mendefinisikan fungsi simbolis murni.

## Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

Untuk menyelesaikan ekspresi sederhana dari satu variabel, kita dapat menggunakan fungsi solve(). Perlu nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode secant.

```
>solve("x^2-2", 1)
```

1.41421356237

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

```
>$&solve(x^2=2, x)
```

$$[x = -\sqrt{2}, x = \sqrt{2}]$$

```
>$&solve(x^2-2, x)
```

$$[x = -\sqrt{2}, x = \sqrt{2}]$$

```
>$&solve(a*x^2+b*x+c=0, x)
```

$$\left[x = \frac{-\sqrt{b^2 - 4ac} - b}{2a}, x = \frac{\sqrt{b^2 - 4ac} - b}{2a}\right]$$

```
>$&solve([a*x+b*y=c, d*x+e*y=f], [x, y])
```

$$\left[\left[x = -\frac{ce}{b(d-5) - ae}, y = \frac{c(d-5)}{b(d-5) - ae}\right]\right]$$

```
>px &= 4*x^8+x^7-x^4-x; $&px
```

$$4x^8 + x^7 - x^4 - x$$

Sekarang kita mencari titik, di mana polinomialnya adalah 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan. Kami menggunakan y=2 dan memeriksa dengan mengevaluasi polinomial pada hasil sebelumnya.

```
>solve(px, 1, y=2), px(%)
```

0.966715594851  
2

Memecahkan ekspresi simbolis dalam bentuk simbolis mengembalikan daftar solusi. Kami menggunakan pemecah simbolik solve() yang disediakan oleh Maxima.

```
>sol &= solve(x^2-x-1,x); $&sol
```

$$\left[ x = \frac{1 - \sqrt{5}}{2}, x = \frac{\sqrt{5} + 1}{2} \right]$$

Cara termudah untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusi secara numerik seperti ekspresi.

```
>longest sol()
```

```
-0.6180339887498949 1.618033988749895
```

Untuk menggunakan solusi secara simbolis dalam ekspresi lain, cara termudah adalah "dengan".

```
>$&x^2 with sol[1], $&expand(x^2-x-1 with sol[2])
```

$$\frac{(\sqrt{5} - 1)^2}{4}$$

```
0
```

Memecahkan sistem persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan solver simbolis solve(). Jawabannya adalah daftar daftar persamaan.

```
>$&solve([x+y=2, x^3+2*y+x=4], [x, y])
```

```
[[x = -1, y = 3], [x = 1, y = 1], [x = 0, y = 2]]
```

Fungsi f() dapat melihat variabel global. Namun seringkali kita ingin menggunakan parameter lokal.

lateks:  $a^x - x^a = 0.1$

dengan a=3.

```
>function f(x,a) := x^a-a^x;
```

Salah satu cara untuk meneruskan parameter tambahan ke f() adalah dengan menggunakan daftar dengan nama fungsi dan parameter (sebaliknya adalah parameter titik koma).

```
>solve({{"f", 3}}, 2, y=0.1)
```

```
Syntax error in expression, or unfinished expression!
```

```
Error in :  
solve({{"f", 3}}, 2, y=0.1)  
^
```

Ini juga bekerja dengan ekspresi. Tapi kemudian, elemen daftar bernama harus digunakan. (Lebih lanjut tentang daftar di tutorial tentang sintaks EMT).

```
>solve({{"x^a-a^x"}, a=3}, 2, y=0.1)
```

Syntax error in expression, or unfinished expression!

```
Error in :  
solve({{"x^a-a^x"}, a=3}, 2, y=0.1)  
^
```

## Menyelesaikan Pertidaksamaan

Untuk menyelesaikan pertidaksamaan, EMT tidak akan dapat melakukannya, melainkan dengan bantuan Maxima, artinya secara eksak (simbolik). Perintah Maxima yang digunakan adalah `fourier_elim()`, yang harus dipanggil dengan

```
>&load(fourier_elim)
```

```
C:/Program Files (x86)/Euler//share/maxima/5.28.0-2/share/fou\  
rier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([x^2 - 1>0], [x]) // x^2-1 > 0
```

$$[1 < x] \vee [x < -1]$$

```
>$&fourier_elim([x^2 - 1<0], [x]) // x^2-1 < 0
```

$$[-1 < x, x < 1]$$

```
>$&fourier_elim([x^2 - 1 # 0], [x]) // x^-1 <> 0
```

$$[-1 < x, x < 1] \vee [1 < x] \vee [x < -1]$$

```
>$&fourier_elim([x # 6], [x])
```

$$[x < 6] \vee [6 < x]$$

```
>$&fourier_elim([x < 1, x > 1],[x]) // tidak memiliki penyelesaian
```

*emptyset*

```
>$&fourier_elim([minf < x, x < inf],[x]) // solusinya R
```

*universalset*

```
>$&fourier_elim([x^3 - 1 > 0],[x])
```

$$[1 < x, x^2 + x + 1 > 0] \vee [x < 1, -x^2 - x - 1 > 0]$$

```
>$&fourier_elim([cos(x) < 1/2],[x]) // ??? gagal
```

$$[1 - 2 \cos x > 0]$$

```
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y]) // sistem pertidaksamaan
```

$$[y - 5 < x, x < y + 7, 10 < y]$$

```
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[y,x])
```

$$[\max(10, x - 7) < y, y < x + 5, 5 < x]$$

```
>$&fourier_elim((x + y < 5) and (x - y > 8), [x,y])
```

$$\left[ y + 8 < x, x < 5 - y, y < -\frac{3}{2} \right]$$

```
>$&fourier_elim(((x + y < 5) and x < 1) or (x - y > 8), [x,y])
```

$$[y + 8 < x] \vee [x < \min(1, 5 - y)]$$

```
>&fourier_elim([max(x,y) > 6, x # 8, abs(y-1) > 12], [x,y])
```

```
[6 < x, x < 8, y < - 11] or [8 < x, y < - 11]
or [x < 8, 13 < y] or [x = y, 13 < y] or [8 < x, x < y, 13 < y]
or [y < x, 13 < y]
```

```
>$&fourier_elim([(x+6)/(x-9) <= 6], [x])
```

$[x = 12] \vee [12 < x] \vee [x < 9]$

## Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, baris dipisahkan dengan titik koma.

```
>A=[1,2;3,4]
```

1	2
3	4

Produk matriks dilambangkan dengan titik.

```
>b=[3;4]
```

3
4

```
>b' // transpose b
```

[3, 4]

```
>inv(A) //inverse A
```

-2	1
1.5	-0.5

```
>A.b //perkalian matriks
```

11
25

```
>A.inv(A)
```

1	0
0	1

Poin utama dari bahasa matriks adalah bahwa semua fungsi dan operator bekerja elemen untuk elemen.

```
>A.A
```

7	10
15	22

```
>A^2 //perpangkatan elemen2 A
```

1	4
9	16

```
>A.A.A
```

37	54
81	118

```
>power(A, 3) //perpangkatan matriks
```

37	54
81	118

```
>A/A //pembagian elemen-elemen matriks yang seletak
```

1	1
1	1

```
>A/b //pembagian elemen2 A oleh elemen2 b kolom demi kolom (karena b vektor kolom)
```

0.333333	0.666667
0.75	1

```
>A\b // hasil kali invers A dan b, A^(-1)b
```

-2
2.5

```
>inv(A).b
```

```
-2  
2.5
```

```
>A\b //A^(-1)A
```

```
1 0  
0 1
```

```
>inv(A).A
```

```
1 0  
0 1
```

```
>A*A //perkalian elemen-elemen matriks seletak
```

```
1 4  
9 16
```

Ini bukan produk matriks, tetapi perkalian elemen demi elemen. Hal yang sama berlaku untuk vektor.

```
>b^2 // perpangkatan elemen-elemen matriks/vektor
```

```
9  
16
```

Jika salah satu operan adalah vektor atau skalar, itu diperluas secara alami.

```
>2*A
```

```
2 4  
6 8
```

Misalnya, jika operan adalah vektor kolom, elemennya diterapkan ke semua baris A.

```
>[1,2]*A
```

```
1 4  
3 8
```

Jika itu adalah vektor baris, itu diterapkan ke semua kolom A.

```
>A*[2,3]
```

```
2 6  
6 12
```

Seseorang dapat membayangkan perkalian ini seolah-olah vektor baris  $v$  telah digandakan untuk membentuk matriks dengan ukuran yang sama dengan  $A$ .

```
>dup([1,2],2) // dup: menduplikasi/menggandakan vektor [1,2] sebanyak 2 kali (baris)
```

1	2
1	2

```
>A*dup([1,2],2)
```

1	4
3	8

Ini juga berlaku untuk dua vektor di mana satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kami menghitung  $i^*j$  untuk  $i,j$  dari 1 hingga 5. Caranya adalah dengan mengalikan 1:5 dengan transposnya. Bahasa matriks Euler secara otomatis menghasilkan tabel nilai.

```
>(1:5)*(1:5)' // hasilkali elemen-elemen vektor baris dan vektor kolom
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Sekali lagi, ingat bahwa ini bukan produk matriks!

```
>(1:5).(1:5)' // hasilkali vektor baris dan vektor kolom
```

55

```
>sum((1:5)*(1:5)) // sama hasilnya
```

55

Bahkan operator seperti `<` atau `==` bekerja dengan cara yang sama.

```
>(1:10)<6 // menguji elemen-elemen yang kurang dari 6
```

[1, 1, 1, 1, 1, 0, 0, 0, 0]

Misalnya, kita dapat menghitung jumlah elemen yang memenuhi kondisi tertentu dengan fungsi `sum()`.

```
>sum((1:10)<6) // banyak elemen yang kurang dari 6
```

5

Euler memiliki operator perbandingan, seperti "`==`", yang memeriksa kesetaraan. Kami mendapatkan vektor 0 dan 1, di mana 1 berarti benar.

```
>t=(1:10)^2; t==25 //menguji elemen2 t yang sama dengan 25 (hanya ada 1)
```

```
[0, 0, 0, 0, 1, 0, 0, 0, 0]
```

Dari vektor seperti itu, "bukan nol" memilih elemen bukan nol.

Dalam hal ini, kami mendapatkan indeks semua elemen lebih besar dari 50.

```
>nonzeros(t>50) //indeks elemen2 t yang lebih besar daripada 50
```

```
[8, 9, 10]
```

Tentu saja, kita dapat menggunakan vektor indeks ini untuk mendapatkan nilai yang sesuai dalam t.

```
>t[nonzeros(t>50)] //elemen2 t yang lebih besar daripada 50
```

```
[64, 81, 100]
```

Sebagai contoh, mari kita cari semua kuadrat dari angka 1 hingga 1000, yaitu 5 modulo 11 dan 3 modulo 13.

```
>t=1:1000; nonzeros(mod(t^2,11)==5 && mod(t^2,13)==3)
```

```
[4, 48, 95, 139, 147, 191, 238, 282, 290, 334, 381, 425, 433, 477, 524, 568, 576, 620, 667, 711, 719, 763, 810, 854, 862, 906, 953, 997]
```

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Ini menggunakan titik mengambang presisi ganda secara internal. Namun, seringkali sangat berguna.

Kita dapat memeriksa keutamaan. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 adalah bilangan prima.

```
>t=1:1000; length(nonzeros(isprime(t^2+1)))
```

```
112
```

Fungsi bukan nol() hanya berfungsi untuk vektor. Untuk matriks, ada mnonzeros().

```
>seed(2); A=random(3,4)
```

0.765761	0.401188	0.406347	0.267829
0.13673	0.390567	0.495975	0.952814
0.548138	0.006085	0.444255	0.539246

Ini mengembalikan indeks elemen, yang bukan nol.

```
>k=mnonzeros(A<0.4) //indeks elemen2 A yang kurang dari 0,4
```

Function mnonzeros not found.  
Try list ... to find functions!

Error in :  
k=mnonzeros(A<0.4) //indeks elemen2 A yang kurang dari 0,4  
^

Indeks ini dapat digunakan untuk mengatur elemen ke beberapa nilai.

```
>mset(A,k,0) //mengganti elemen2 suatu matriks pada indeks tertentu
```

Function mset not found.  
Try list ... to find functions!

Error in :  
mset(A,k,0) //mengganti elemen2 suatu matriks pada indeks tertentu  
^

Fungsi mset() juga dapat mengatur elemen pada indeks ke entri dari beberapa matriks lainnya.

```
>mset(A,k,-random(size(A)))
```

Function mset not found.  
Try list ... to find functions!

Error in :  
mset(A,k,-random(size(A)))  
^

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

```
>mget(A,k)
```

Function mget not found.  
Try list ... to find functions!

Error in :  
mget(A,k)  
^

Fungsi lain yang berguna adalah ekstrem, yang mengembalikan nilai minimal dan maksimal di setiap baris matriks dan posisinya.

```
>ex=extrema(A)
```

0.267829	4	0.765761	1
0.13673	1	0.952814	4
0.006085	2	0.548138	1

Kita dapat menggunakan ini untuk mengekstrak nilai maksimal di setiap baris.

```
>ex[,3]'
```

```
[0.765761, 0.952814, 0.548138]
```

Ini, tentu saja, sama dengan fungsi max().

```
>max(A)'
```

```
[0.765761, 0.952814, 0.548138]
```

Tetapi dengan mget(), kita dapat mengekstrak indeks dan menggunakan informasi ini untuk mengekstrak elemen pada posisi yang sama dari matriks lain.

```
>j=(1:rows(A))' | ex[,4], mget(-A, j)
```

```
1           1  
2           4  
3           1
```

```
Function mget not found.  
Try list ... to find functions!
```

```
Error in :  
j=(1:rows(A))' | ex[,4], mget(-A, j)  
^
```

## Fungsi Matriks Lainnya (Membangun Matriks)

Untuk membangun matriks, kita dapat menumpuk satu matriks di atas yang lain. Jika keduanya tidak memiliki jumlah kolom yang sama, kolom yang lebih pendek akan diisi dengan 0.

```
>v=1:3; v_v
```

```
1           2           3  
1           2           3
```

Demikian juga, kita dapat melampirkan matriks ke yang lain secara berdampingan, jika keduanya memiliki jumlah baris yang sama.

```
>A=random(3,4); A|v'
```

0.032444	0.0534171	0.595713	0.564454	1
0.83916	0.175552	0.396988	0.83514	2
0.0257573	0.658585	0.629832	0.770895	3

Jika mereka tidak memiliki jumlah baris yang sama, matriks yang lebih pendek diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan real yang dilampirkan pada matriks akan digunakan sebagai kolom yang diisi dengan bilangan real tersebut.

```
>A | 1
```

0.032444	0.0534171	0.595713	0.564454	1
0.83916	0.175552	0.396988	0.83514	1
0.0257573	0.658585	0.629832	0.770895	1

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

```
> [v;v]
```

1	2	3
1	2	3

```
> [v',v']
```

Illegal matrix element

Error in :  
[v',v']  
^

Tujuan utama dari ini adalah untuk menafsirkan vektor ekspresi untuk vektor kolom.

```
>"[x,x^2]"(v')
```

Illegal matrix element  
Error in expression: [x,x^2]

Error in :  
"[x,x^2]"(v')  
^

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

```
>C=zeros(2,4); rows(C), cols(C), size(C), length(C)
```

2  
4  
[2, 4]  
4

Untuk vektor, ada panjang().

```
>length(2:10)
```

9

Ada banyak fungsi lain, yang menghasilkan matriks.

```
>ones(2,2)
```

```
1           1  
1           1
```

Ini juga dapat digunakan dengan satu parameter. Untuk mendapatkan vektor dengan angka selain 1, gunakan yang berikut ini.

```
>ones(5)*6
```

```
[6,   6,   6,   6,   6]
```

Juga matriks bilangan acak dapat dihasilkan dengan acak (distribusi seragam) atau normal (distribusi Gau).

```
>random(2,2)
```

```
0.66566      0.831835  
0.977        0.544258
```

Berikut adalah fungsi lain yang berguna, yang merestrukturisasi elemen matriks menjadi matriks lain.

```
>redim(1:9,3,3) // menyusun elemen 1, 2, 3, ..., 9 ke bentuk matriks 3x3
```

```
1           2           3  
4           5           6  
7           8           9
```

Dengan fungsi berikut, kita dapat menggunakan ini dan fungsi dup untuk menulis fungsi rep(), yang mengulang vektor n kali.

```
>function rep(v,n) := redim(dup(v,n),1,n*cols(v))
```

Mari kita uji.

```
>rep(1:3,5)
```

```
[1,   2,   3,   1,   2,   3,   1,   2,   3,   1,   2,   3]
```

Fungsi multdup() menduplikasi elemen vektor.

```
>multdup(1:3,5), multdup(1:3,[2,3,2])
```

```
[1,   1,   1,   1,   1,   2,   2,   2,   2,   2,   3,   3,   3,   3]  
[1,   1,   2,   2,   2,   3,   3]
```

Fungsi flipx() dan flipy() mengembalikan urutan baris atau kolom matriks. Yaitu, fungsi flipx() membalik secara horizontal.

```
>flipx(1:5) //membalik elemen2 vektor baris
```

```
[5, 4, 3, 2, 1]
```

Untuk rotasi, Euler memiliki `rotleft()` dan `rotright()`.

```
>rotleft(1:5) // memutar elemen2 vektor baris
```

```
[2, 3, 4, 5, 1]
```

Sebuah fungsi khusus adalah `drop(v,i)`, yang menghilangkan elemen dengan indeks di  $i$  dari vektor  $v$ .

```
>drop(10:20,3)
```

```
[10, 11, 13, 14, 15, 16, 17, 18, 19, 20]
```

Perhatikan bahwa vektor  $i$  di `drop(v,i)` mengacu pada indeks elemen di  $v$ , bukan nilai elemen. Jika Anda ingin menghapus elemen, Anda harus menemukan elemennya terlebih dahulu. Fungsi `indexof(v,x)` dapat digunakan untuk mencari elemen  $x$  dalam vektor terurut  $v$ .

```
>v=primes(50), i=indexof(v,10:20), drop(v,i)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
[0, 5, 0, 6, 0, 0, 0, 7, 0, 8, 0]
[2, 3, 5, 7, 23, 29, 31, 37, 41, 43, 47]
```

Seperti yang Anda lihat, tidak ada salahnya untuk memasukkan indeks di luar rentang (seperti 0), indeks ganda, atau indeks yang tidak diurutkan.

```
>drop(1:10,shuffle([0,0,5,5,7,12,12]))
```

```
[1, 2, 3, 4, 6, 8, 9, 10]
```

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal. Kita mulai dengan matriks identitas.

```
>A=id(5) // matriks identitas 5x5
```

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Kemudian kita atur diagonal bawah (-1) menjadi 1:4.

```
>setdiag(A,-1,1:4) //mengganti diagonal di bawah diagonal utama
```

1	0	0	0	0
1	1	0	0	0
0	2	1	0	0
0	0	3	1	0
0	0	0	4	1

Perhatikan bahwa kami tidak mengubah matriks A. Kami mendapatkan matriks baru sebagai hasil dari setdiag().

Berikut adalah fungsi, yang mengembalikan matriks tri-diagonal.

```
>function tridiag (n,a,b,c) := setdiag(setdiag(b*id(n),1,c),-1,a); ...
>tridiag(5,1,2,3)
```

2	3	0	0	0
1	2	3	0	0
0	1	2	3	0
0	0	1	2	3
0	0	0	1	2

Diagonal suatu matriks juga dapat diekstraksi dari matriks tersebut. Untuk mendemonstrasikan ini, kami merestrukturisasi vektor 1:9 menjadi matriks 3x3.

```
>A=redim(1:9, 3, 3)
```

1	2	3
4	5	6
7	8	9

Sekarang kita dapat mengekstrak diagonal.

```
>d=getdiag(A, 0)
```

```
Function getdiag not found.  
Try list ... to find functions!
```

```
Error in :  
d=getdiag(A, 0)  
^
```

Misalnya. Kita dapat membagi matriks dengan diagonalnya. Bahasa matriks memperhatikan bahwa vektor kolom d diterapkan ke matriks baris demi baris.

```
>fraction A/d'
```

```

Variable d not found!
fraction:   f=getformat();

Error in :
fraction A/d'
^

```

## Vektorisasi

---

Hampir semua fungsi di Euler juga berfungsi untuk input matriks dan vektor, kapan pun ini masuk akal. Misalnya, fungsi `sqrt()` menghitung akar kuadrat dari semua elemen vektor atau matriks.

```

>sqrt(1:3)

[1, 1.41421, 1.73205]

```

Jadi Anda dapat dengan mudah membuat tabel nilai. Ini adalah salah satu cara untuk memplot suatu fungsi (alternatifnya menggunakan ekspresi).

```

>x=1:0.01:5; y=log(x)/x^2; // terlalu panjang untuk ditampilkan

```

Dengan ini dan operator titik dua `a:delta:b`, vektor nilai fungsi dapat dihasilkan dengan mudah. Pada contoh berikut, kita membangkitkan vektor nilai `t[i]` dengan spasi 0,1 dari -1 hingga 1. Kemudian kita membangkitkan vektor nilai fungsi

lateks:  $s = t^3 - t$

```

>t=-1:0.1:1; s=t^3-t

```

```

[0, 0.171, 0.288, 0.357, 0.384, 0.375, 0.336, 0.273, 0.192,
0.099, 0, -0.099, -0.192, -0.273, -0.336, -0.375, -0.384,
-0.357, -0.288, -0.171, 0]

```

EMT memperluas operator untuk skalar, vektor, dan matriks dengan cara yang jelas.

Misalnya, vektor kolom dikalikan vektor baris menjadi matriks, jika operator diterapkan. Berikut ini,  $v'$  adalah vektor yang ditransposisikan (vektor kolom).

```

>shortest (1:5)*(1:5)'

```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Perhatikan, bahwa ini sangat berbeda dari produk matriks. Produk matriks dilambangkan dengan titik `"."` di EMT.

```

>(1:5).(1:5)'

```

Secara default, vektor baris dicetak dalam format yang ringkas.

```
>[1,2,3,4]
```

```
[1, 2, 3, 4]
```

Untuk matriks operator khusus . menunjukkan perkalian matriks, dan A' menunjukkan transpos. Matriks 1x1 dapat digunakan seperti bilangan real.

```
>v:=[1,2]; v.v', %^2
```

```
5  
25
```

Untuk mentranspos matriks kita menggunakan apostrof.

```
>v=1:4; v'
```

```
1  
2  
3  
4
```

Jadi kita dapat menghitung matriks A kali vektor b.

```
>A=[1,2,3,4;5,6,7,8]; A.v'
```

```
30  
70
```

Perhatikan bahwa v masih merupakan vektor baris. Jadi v'.v berbeda dari v.v'.

```
>v'.v
```

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

v.v' menghitung norma v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang bekerja seperti bilangan real.

```
>v.v'
```

```
30
```

Ada juga fungsi norma (bersama dengan banyak fungsi lain dari Aljabar Linier).

```
>norm(v)^2
```

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan aturannya.

- Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap elemen.
- Operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan berpasangan ke elemen matriks.
- Jika kedua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.

Misalnya, nilai skalar kali vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks kali vektor (dengan \*, bukan .) memperluas vektor ke ukuran matriks dengan menduplikasinya.

Berikut ini adalah kasus sederhana dengan operator ·

```
> [1,2,3]^2
```

[1, 4, 9]

Berikut adalah kasus yang lebih rumit. Vektor baris dikalikan dengan vektor kolom mengembang keduanya dengan menduplikasi.

```
>v:=[1,2,3]; v*v'
```

1	2	3
2	4	6
3	6	9

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan \*!

```
>v.v'
```

14

Ada banyak fungsi matriks. Kami memberikan daftar singkat. Anda harus berkonsultasi dengan dokumentasi untuk informasi lebih lanjut tentang perintah ini.

sum,prod menghitung jumlah dan produk dari baris cumsum,cumprod melakukan hal yang sama secara kumulatif menghitung nilai ekstrem dari setiap baris extrema mengembalikan vektor dengan informasi ekstrim diag(A,i) mengembalikan diagonal ke-i setdiag(A,i,v) mengatur diagonal ke-i id(n) matriks identitas det(A) penentu charpoly(A) polinomial karakteristik nilai eigen(A) nilai eigen

```
>v*v, sum(v*v), cumsum(v*v)
```

[1, 4, 9]  
14  
[1, 5, 14]

Operator : menghasilkan vektor baris spasi yang sama, opsional dengan ukuran langkah.

```
>1:4, 1:2:10
```

```
[1, 2, 3, 4]  
[1, 3, 5, 7, 9]
```

Untuk menggabungkan matriks dan vektor ada operator "|" dan "».

```
>[1,2,3] | [4,5], [1,2,3]_1
```

```
[1, 2, 3, 4, 5]  
1 2 3  
1 1 1
```

Unsur-unsur matriks disebut dengan "A[i,j]".

```
>A:=[1,2,3;4,5,6;7,8,9]; A[2,3]
```

```
6
```

Untuk vektor baris atau kolom, v[i] adalah elemen ke-i dari vektor. Untuk matriks, ini mengembalikan baris ke-i lengkap dari matriks.

```
>v:=[2,4,6,8]; v[3], A[3]
```

```
6  
[7, 8, 9]
```

Indeks juga bisa menjadi vektor baris dari indeks. : menunjukkan semua indeks.

```
>v[1:2], A[:,2]
```

```
[2, 4]  
2  
5  
8
```

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

```
>A[,2:3]
```

```
2 3  
5 6  
8 9
```

Untuk tujuan vektorisasi, elemen matriks dapat diakses seolah-olah mereka adalah vektor.

```
>A{4}
```

```
4
```

Matriks juga dapat diratakan, menggunakan fungsi `redim()`. Ini diimplementasikan dalam fungsi `flatten()`.

```
>redim(A,1,prod(size(A))), flatten(A)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Untuk menggunakan matriks untuk tabel, mari kita reset ke format default, dan menghitung tabel nilai sinus dan kosinus. Perhatikan bahwa sudut dalam radian secara default.

```
>defformat; w=0°:45°:360°; w=w'; deg(w)
```

```
0
45
90
135
180
225
270
315
360
```

Sekarang kita menambahkan kolom ke matriks.

```
>M = deg(w)|w|cos(w)|sin(w)
```

0	0	1	0
45	0.785398	0.707107	0.707107
90	1.5708	0	1
135	2.35619	-0.707107	0.707107
180	3.14159	-1	0
225	3.92699	-0.707107	-0.707107
270	4.71239	0	-1
315	5.49779	0.707107	-0.707107
360	6.28319	1	0

Dengan menggunakan bahasa matriks, kita dapat menghasilkan beberapa tabel dari beberapa fungsi sekali-gus.

Dalam contoh berikut, kita menghitung  $t[j]^i$  untuk  $i$  dari 1 hingga  $n$ . Kami mendapatkan matriks, dimana setiap baris adalah sebuah vektor fungsi  $a_{\{i, j\}} = t_j^i$ ,  $\quad 1 \leq j \leq 101, \quad 1 \leq i \leq n$

Fungsi yang tidak berfungsi untuk input vektor harus "divektorkan". Ini dapat dicapai dengan kata kunci "peta" dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.

Integrasi numerik `terintegrasi()` hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektor.

```
>function map f(x) := integrate("x^x", 1, x)
```

Kata kunci "peta" membuat vektor fungsi. Fungsinya sekarang akan bekerja untuk vektor bilangan.

```
>f([1:5])
```

```
[0, 2.05045, 13.7251, 113.336, 1241.03]
```

## Sub-Matriks dan Matriks-Elemen

Untuk mengakses elemen matriks, gunakan notasi braket.

```
>A=[1,2,3;4,5,6;7,8,9], A[2,2]
```

1	2	3
4	5	6
7	8	9
5		

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[2]
```

```
[4, 5, 6]
```

Dalam kasus vektor baris atau kolom, ini mengembalikan elemen vektor.

```
>v=1:3; v[2]
```

```
2
```

Untuk memastikan, Anda mendapatkan baris pertama untuk matriks  $1 \times n$  dan  $m \times n$ , tentukan semua kolom menggunakan indeks kedua kosong.

```
>A[2, ]
```

```
[4, 5, 6]
```

Jika indeks adalah vektor indeks, Euler akan mengembalikan baris matriks yang sesuai.

Di sini kita ingin baris pertama dan kedua dari A.

```
>A[[1,2]]
```

1	2	3
4	5	6

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Tepatnya, kami tidak mengubah A di sini, tetapi menghitung versi A yang disusun ulang.

```
>A[[3,2,1]]
```

7	8	9
4	5	6
1	2	3

Trik indeks bekerja dengan kolom juga.

Contoh ini memilih semua baris A dan kolom kedua dan ketiga.

```
>A[1:3,2:3]
```

2	3
5	6
8	9

Untuk singkatan ":" menunjukkan semua indeks baris atau kolom.

```
>A[:,3]
```

3
6
9

Atau, biarkan indeks pertama kosong.

```
>A[,2:3]
```

2	3
5	6
8	9

Kita juga bisa mendapatkan baris terakhir dari A.

```
>A[-1]
```

[7, 8, 9]
-----------

Sekarang mari kita ubah elemen A dengan menetapkan submatriks A ke beberapa nilai. Ini sebenarnya mengubah matriks A yang disimpan.

```
>A[1,1]=4
```

4	2	3
4	5	6
7	8	9

Kami juga dapat menetapkan nilai ke baris A.

```
>A[1]=[-1,-1,-1]
```

-1	-1	-1
4	5	6
7	8	9

Kami bahkan dapat menetapkan sub-matriks jika memiliki ukuran yang tepat.

```
>A[1:2,1:2]=[5,6;7,8]
```

5	6	-1
7	8	6
7	8	9

Selain itu, beberapa jalan pintas diperbolehkan.

```
>A[1:2,1:2]=0
```

0	0	-1
0	0	6
7	8	9

Peringatan: Indeks di luar batas mengembalikan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Ingat, bagaimanapun, bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

```
>A[4]
```

Index 4 out of bounds!

Error in :

A[4]

^

## Menyortir dan Mengacak

Fungsi sort() mengurutkan vektor baris.

```
>sort([5,6,4,8,1,9])
```

[1, 4, 5, 6, 8, 9]

Seringkali perlu untuk mengetahui indeks dari vektor yang diurutkan dalam vektor aslinya. Ini dapat digunakan untuk menyusun ulang vektor lain dengan cara yang sama.

Mari kita mengocok vektor.

```
>v=shuffle(1:10)
```

[4, 5, 10, 6, 8, 9, 1, 7, 2, 3]

Indeks berisi urutan yang tepat dari v.

```
>{vs,ind}=sort(v); v[ind]
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Ini bekerja untuk vektor string juga.

```
>s=[ "a", "d", "e", "a", "aa", "e"]
```

```
a  
d  
e  
a  
aa  
e
```

```
>{ss,ind}=sort(s); ss
```

```
a  
a  
aa  
d  
e  
e
```

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

```
[4, 1, 5, 2, 6, 3]
```

Fungsi unik mengembalikan daftar elemen unik vektor yang diurutkan.

```
>intrandom(1,10,10), unique(%)
```

```
[4, 4, 9, 2, 6, 5, 10, 6, 5, 1]  
[1, 2, 4, 5, 6, 9, 10]
```

Ini bekerja untuk vektor string juga.

```
>unique(s)
```

```
a  
aa  
d  
e
```

## Aljabar linier

EMT memiliki banyak fungsi untuk menyelesaikan sistem linier, sistem sparse, atau masalah regresi.

Untuk sistem linier  $Ax=b$ , Anda dapat menggunakan algoritma Gauss, matriks invers atau kecocokan linier. Operator  $A\b$  menggunakan versi algoritma Gauss.

```
>A=[1,2;3,4]; b=[5;6]; A\b
```

```
-4  
4.5
```

Untuk contoh lain, kami membuat matriks 200x200 dan jumlah barisnya. Kemudian kita selesaikan  $Ax=b$  menggunakan matriks invers. Kami mengukur kesalahan sebagai deviasi maksimal semua elemen dari 1, yang tentu saja merupakan solusi yang benar.

```
>A=normal(200,200); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

```
6.59472476627343e-013
```

Jika sistem tidak memiliki solusi, kecocokan linier meminimalkan norma kesalahan  $Ax-b$ .

```
>A=[1,2,3;4,5,6;7,8,9]
```

1	2	3
4	5	6
7	8	9

Determinan matriks ini adalah 0.

```
>det (A)
```

```
0
```

## Matriks Simbolik

Maxima memiliki matriks simbolis. Tentu saja, Maxima dapat digunakan untuk masalah aljabar linier se-derhana seperti itu. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan :=, dan kemudian menggunakannya dalam ekspresi simbolis. Bentuk [...] biasa untuk mendefinisikan matriks dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
```

$$\begin{pmatrix} a & 1 & 1 \\ 1 & a & 1 \\ 1 & 1 & a \end{pmatrix}$$

```
>$&det (A), $&factor (%)
```

$$a (a^2 - 1) - 2 a + 2$$

$$(a - 1)^2 (a + 2)$$

```
> $& invert(A) with a=0
```

$$\begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

```
> A &= [1,a;b,2]; $A
```

$$\begin{pmatrix} 1 & a \\ b & 2 \end{pmatrix}$$

Seperi semua variabel simbolik, matriks ini dapat digunakan dalam ekspresi simbolik lainnya.

```
> $& det(A-x*ident(2)), $& solve(% ,x)
```

$$(1 - x) (2 - x) - a b$$

$$\left[ x = \frac{3 - \sqrt{4 a b + 1}}{2}, x = \frac{\sqrt{4 a b + 1} + 3}{2} \right]$$

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah vektor dengan dua vektor nilai eigen dan multiplisitas.

```
> $& eigenvalues([a,1;1,a])
```

$$[[a - 1, a + 1], [1, 1]]$$

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

```
> $& eigenvectors([a,1;1,a]), &%[2][1][1]
```

$$[[[a - 1, a + 1], [1, 1]], [[[1, -1]], [[1, 1]]]]$$

$$[1, -1]$$

Matriks simbolik dapat dievaluasi dalam Euler secara numerik seperti ekspresi simbolik lainnya.

```
> A(a=4,b=5)
```

$$\begin{matrix} 1 & 4 \\ 5 & 2 \end{matrix}$$

Dalam ekspresi simbolik, gunakan dengan.

```
> $&A with [a=4, b=5]
```

$$\begin{pmatrix} 1 & 4 \\ 5 & 2 \end{pmatrix}$$

Akses ke baris matriks simbolik bekerja seperti halnya dengan matriks numerik.

```
> $&A[1]
```

$$[1, a]$$

Ekspresi simbolis dapat berisi tugas. Dan itu mengubah matriks A.

```
> &A[1,1]:=t+1; $&A
```

$$\begin{pmatrix} t+1 & a \\ b & 2 \end{pmatrix}$$

Ada fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk ini, lihat dokumentasi Maxima atau tutorial tentang Maxima di EMT.

```
> v &:= makelist(1/(i+j), i, 1, 3); $v
```

$$\left[ \frac{1}{j+1}, \frac{1}{j+2}, \frac{1}{j+3} \right]$$

```
> B &:= [1, 2; 3, 4]; $B, $&invert(B)
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

Hasilnya dapat dievaluasi secara numerik dalam Euler. Untuk informasi lebih lanjut tentang Maxima, lihat pengantar Maxima.

```
> $&invert(B)()
```

$$\begin{array}{cc} -2 & 1 \\ 1.5 & -0.5 \end{array}$$

Euler juga memiliki fungsi `xinv()` yang kuat, yang membuat upaya lebih besar dan mendapatkan hasil yang lebih tepat.

Perhatikan, bahwa dengan `:=` matriks `B` telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita bisa menggunakannya di sini.

```
>longest B.xinv(B)
```

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

Misalnya. nilai eigen dari `A` dapat dihitung secara numerik.

```
>A=[1,2,3;4,5,6;7,8,9]; real(eigenvalues(A))
```

$$[16.1168, -1.11684, 0]$$

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

```
>${&eigenvalues (@A)}
```

$$\left[ \left[ \frac{15 - 3\sqrt{33}}{2}, \frac{3\sqrt{33} + 15}{2}, 0 \right], [1, 1, 1] \right]$$

## Nilai Numerik dalam Ekspresi simbolis

Ekspresi simbolis hanyalah string yang berisi ekspresi. Jika kita ingin mendefinisikan nilai baik untuk ekspresi simbolik maupun ekspresi numerik, kita harus menggunakan "`:=`".

```
>A &:= [1,pi;4,5]
```

$$\begin{matrix} 1 & 3.14159 \\ 4 & 5 \end{matrix}$$

Masih ada perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolis, pendekatan fraksional untuk real akan digunakan.

```
>${&A}
```

$$\begin{pmatrix} 1 & \frac{245850922}{78256779} \\ 4 & 5 \end{pmatrix}$$

Untuk menghindarinya, ada fungsi "m xmset(variable)".

```
>m xmset (A); ${&A}
```

$$\begin{pmatrix} 1 & 3.141592653589793 \\ 4 & 5 \end{pmatrix}$$

Maxima juga dapat menghitung dengan angka floating point, dan bahkan dengan angka floating besar dengan 32 digit. Namun, evaluasinya jauh lebih lambat.

```
>${bf}loat(sqrt(2)), ${float(sqrt(2))}
```

$$1.4142135623730950488016887242097_B \times 10^0$$

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah.

```
>fpprec:=100; ${bf}loat(pi)
```

$$3.14159265358979323846264338327950288419716939937510582097494\\ 4592307816406286208998628034825342117068b0$$

Variabel numerik dapat digunakan dalam ekspresi simbolis apa pun menggunakan "@var".

Perhatikan bahwa ini hanya diperlukan, jika variabel telah didefinisikan dengan ":=" atau "=" sebagai variabel numerik.

```
>B:=[1,pi;3,4]; ${det}@B
```

$$-5.424777960769379$$

## Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk perhitungan suku bunga. Kami melakukannya secara numerik dan simbolis untuk menunjukkan kepada Anda bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata.

Asumsikan Anda memiliki modal awal 5000 (katakanlah dalam dolar).

```
>K=5000
```

$$5000$$

Sekarang kita asumsikan tingkat bunga 3% tambahkan satu tarif sederhana dan hitung hasilnya.

```
>K*1.03
```

$$5150$$

Euler akan memahami sintaks berikut juga.

```
>K+K*3%
```

$$5150$$

Tetapi lebih mudah menggunakan faktornya

```
>q=1+3%, K*q
```

```
1.03  
5150
```

Selama 10 tahun, kita cukup mengalikan faktornya dan mendapatkan nilai akhir dengan suku bunga majemuk.

```
>K*q^10
```

```
6719.58189672
```

Untuk tujuan kita, kita dapat mengatur format menjadi 2 digit setelah titik desimal.

```
>format(12,2); K*q^10
```

```
6719.58
```

Mari kita cetak yang dibulatkan menjadi 2 digit dalam kalimat lengkap.

```
>"Starting from " + K + "$ you get " + round(K*q^10,2) + "$."
```

```
Starting from 5000$ you get 6719.58$.
```

Bagaimana jika kita ingin mengetahui hasil antara dari tahun 1 sampai tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak harus menulis loop, tetapi cukup masukkan

```
>K*q^(0:10)
```

```
Column 1 to 5:  
5000.00 5150.00 5304.50 5463.64 5627.54  
Column 6 to 10:  
5796.37 5970.26 6149.37 6333.85 6523.87  
Column 11:  
6719.58
```

Bagaimana keajaiban ini bekerja? Pertama ekspresi 0:10 mengembalikan vektor bilangan bulat.

```
>short 0:10
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Kemudian semua operator dan fungsi dalam Euler dapat diterapkan pada elemen vektor untuk elemen. Jadi

```
>short q^(0:10)
```

```
[1, 1.03, 1.0609, 1.0927, 1.1255, 1.1593, 1.1941, 1.2299,  
1.2668, 1.3048, 1.3439]
```

adalah vektor faktor  $q^0$  sampai  $q^{10}$ . Ini dikalikan dengan  $K$ , dan kami mendapatkan vektor nilai.

```
>VK=K*q^(0:10);
```

Tentu saja, cara realistik untuk menghitung suku bunga ini adalah dengan membulatkan ke sen terdekat setelah setiap tahun. Mari kita tambahkan fungsi untuk ini.

```
>function oneyear (K) := round(K*q, 2)
```

Mari kita bandingkan dua hasil, dengan dan tanpa pembulatan.

```
>longest oneyear(1234.57), longest 1234.57*q
```

```
1271.61  
1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke- $n$ , dan kita harus mengulang selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.

Cara termudah adalah iterasi fungsi, yang mengulangi fungsi tertentu beberapa kali.

```
>VKr=iterate("oneyear",5000,10)
```

```
Column 1 to 5:  
5000.00 5150.00 5304.50 5463.64 5627.55  
Column 6 to 10:  
5796.38 5970.27 6149.38 6333.86 6523.88  
Column 11:  
6719.60
```

Kami dapat mencetaknya dengan cara yang ramah, menggunakan format kami dengan tempat desimal tetap.

```
>VKr'
```

```
5000.00  
5150.00  
5304.50  
5463.64  
5627.55  
5796.38  
5970.27  
6149.38  
6333.86  
6523.88  
6719.60
```

Untuk mendapatkan elemen tertentu dari vektor, kami menggunakan indeks dalam tanda kurung siku.

```
>VKr[2], VKr[1:3]
```

```
5150.00  
5000.00 5150.00 5304.50
```

Anehnya, kita juga bisa menggunakan vektor indeks. Ingat bahwa 1:3 menghasilkan vektor [1,2,3]. Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

```
>VKr[-1], VK[-1]
```

```
6719.60  
6719.58
```

Perbedaannya sangat kecil.

## Memecahkan Persamaan

Sekarang kita mengambil fungsi yang lebih maju, yang menambahkan tingkat uang tertentu setiap tahun.

```
>function onepay (K) := K*q+R
```

Kita tidak perlu menentukan q atau R untuk definisi fungsi. Hanya jika kita menjalankan perintah, kita harus mendefinisikan nilai-nilai ini. Kami memilih R=200.

```
>R=200; iterate("onepay",5000,10)
```

```
Column 1 to 5:  
5000.00 5350.00 5710.50 6081.82 6464.27  
Column 6 to 10:  
6858.20 7263.94 7681.86 8112.32 8555.69  
Column 11:  
9012.36
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",5000,10)
```

```
Column 1 to 5:  
5000.00 4950.00 4898.50 4845.45 4790.82  
Column 6 to 10:  
4734.54 4676.58 4616.88 4555.38 4492.04  
Column 11:  
4426.81
```

Kami melihat bahwa uang berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama, tetapi menghapus 200, kita kehilangan uang setiap tahun.

Bagaimana kita bisa menentukan berapa tahun uang itu akan bertahan? Kita harus menulis loop untuk ini. Cara termudah adalah dengan iterasi cukup lama.

```
>VKR=iterate("onepay",5000,50)
```

```
Column 1 to 5:  
    5000.00    4950.00    4898.50    4845.45    4790.82  
Column 6 to 10:  
    4734.54    4676.58    4616.88    4555.38    4492.04  
Column 11 to 15:  
    4426.81    4359.61    4290.40    4219.11    4145.68  
Column 16 to 20:  
    4070.05    3992.16    3911.92    3829.28    3744.16  
Column 21 to 25:  
    3656.48    3566.18    3473.16    3377.36    3278.68  
Column 26 to 30:  
    3177.04    3072.35    2964.52    2853.45    2739.06  
Column 31 to 35:  
    2621.23    2499.87    2374.86    2246.11    2113.49  
Column 36 to 40:  
    1976.90    1836.20    1691.29    1542.03    1388.29  
Column 41 to 45:  
    1229.94    1066.84    898.84     725.81     547.58  
Column 46 to 50:  
    364.01     174.93    -19.83    -220.42    -427.03  
Column 51:  
    -639.84
```

Dengan menggunakan bahasa matriks, kita dapat menentukan nilai negatif pertama dengan cara berikut.

```
>min(nonzeros(VKR<0) )
```

```
48.00
```

Alasan untuk ini adalah bahwa bukan nol(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.

Karena vektor selalu dimulai dengan indeks 1, jawabannya adalah 47 tahun.

Fungsi iterate() memiliki satu trik lagi. Itu bisa mengambil kondisi akhir sebagai argumen. Kemudian akan mengembalikan nilai dan jumlah iterasi.

```
>{x,n}=iterate("onepay",5000,till="x<0"); x, n,
```

```
-19.83  
47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Apa yang akan menjadi tingkat bunga?

Ini adalah pertanyaan yang hanya bisa dijawab dengan angka. Di bawah ini, kita akan mendapatkan formula yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada formula yang mudah untuk tingkat bunga. Tapi untuk saat ini, kami bertujuan untuk solusi numerik.

Langkah pertama adalah mendefinisikan fungsi yang melakukan iterasi sebanyak n kali. Kami menambahkan semua parameter ke fungsi ini.

```
>function f(K,R,P,n) := iterate("x*(1+P/100)+R",K,n;P,R)[-1]
```

Iterasinya sama seperti di atas

$$x_{n+1} = x_n \cdot \left(1 + \frac{P}{100}\right) + R$$

Tapi kami tidak lagi menggunakan nilai global R dalam ekspresi kami. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

Selain itu, kami hanya tertarik pada nilai terakhir. Jadi kita ambil indeks [-1].

Mari kita coba tes.

```
>f(5000,-200,3,47)
```

-19.83

Sekarang kita bisa menyelesaikan masalah kita.

```
>solve("f(5000,-200,x,50)",3)
```

```
Floating point error!
secant:           x2=x1-y1*(x1-x0)/(y1-y0);
solve: if (isvar("eps")) then return secant(f$,a,b,y,args(),eps=eps ...
```

Rutin memecahkan memecahkan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15Fungsi solve() selalu membutuhkan nilai awal.

Kita dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang dapat kita keluarkan per tahun sehingga modal awal habis setelah 20 tahun dengan asumsi tingkat bunga 3tahun.

```
>solve("f(5000,x,3,20)",-200)
```

```
Floating point error!
secant:           x2=x1-y1*(x1-x0)/(y1-y0);
solve: if (isvar("eps")) then return secant(f$,a,b,y,args(),eps=eps ...
```

Perhatikan bahwa Anda tidak dapat memecahkan jumlah tahun, karena fungsi kami mengasumsikan n sebagai nilai integer.

### **Solusi Simbolik untuk Masalah Suku Bunga**

Kita dapat menggunakan bagian simbolik dari Euler untuk mempelajari masalah tersebut. Pertama kita mendefinisikan fungsi onepay() kita secara simbolis.

```
>function op(K) &= K*q+R; $&op(K)
```

$$R + qK$$

Kita sekarang dapat mengulangi ini.

```
> $&op(op(op(op(K)))) , $&expand(%)
```

$$q (q (q (R + q K) + R) + R) + R$$

$$q^3 R + q^2 R + q R + R + q^4 K$$

Kami melihat sebuah pola. Setelah n periode yang kita miliki  
lateks:  $K_n = q^n K + R(1 + q + \dots + q^{n-1}) = q^n K + \frac{q^n - 1}{q - 1} R$   
Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

```
>&sum(q^k, k, 0, n-1); $& % = ev(%, simpsum)
```

$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}$$

Ini agak rumit. Jumlahnya dievaluasi dengan bendera "simpsum" untuk menguranginya menjadi hasil bagi.  
Mari kita membuat fungsi untuk ini.

```
>function fs(K, R, P, n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K, R, P, n)
```

$$\frac{100 \left(\left(\frac{P}{100} + 1\right)^n - 1\right) R}{P} + K \left(\frac{P}{100} + 1\right)^n$$

Fungsi tersebut melakukan hal yang sama seperti fungsi f kita sebelumnya. Tapi itu lebih efektif.

```
>longest f(5000, -200, 3, 47), longest fs(5000, -200, 3, 47)
```

```
-19.82504734650985  
-19.82504734652684
```

Kita sekarang dapat menggunakan untuk menanyakan waktu n. Kapan modal kita habis? Dugaan awal kami adalah 30 tahun.

```
>solve("fs(5000, -330, 3, x)", 30)
```

```
20.51
```

Jawaban ini mengatakan bahwa itu akan menjadi negatif setelah 21 tahun.

Kita juga dapat menggunakan sisi simbolis Euler untuk menghitung formula pembayaran.

Asumsikan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) meninggalkan sisa hutang sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas

```
>equ &= fs(K, R, P, n)=Kn; $&equ
```

$$\frac{100 \left( \left( \frac{P}{100} + 1 \right)^n - 1 \right) R}{P} + K \left( \frac{P}{100} + 1 \right)^n = Kn$$

Biasanya rumus ini diberikan dalam bentuk

$$i = \frac{P}{100}$$

```
>equ &= (equ with P=100*i); $&equ
```

$$\frac{\left( (i+1)^n - 1 \right) R}{i} + (i+1)^n K = Kn$$

Kita dapat memecahkan tingkat R secara simbolis.

```
>$&solve(equ, R)
```

$$\left[ R = \frac{i Kn - i (i+1)^n K}{(i+1)^n - 1} \right]$$

Seperti yang Anda lihat dari rumus, fungsi ini mengembalikan kesalahan titik mengambang untuk  $i=0$ . Euler tetap merencanakannya.

Tentu saja, kami memiliki batas berikut.

```
>$&limit(R(5000, 0, x, 10), x, 0)
```

$$\lim_{x \rightarrow 0} R(5000, 0, x, 10)$$

Jelas, tanpa bunga kita harus membayar kembali 10 tarif 500.

Persamaan juga dapat diselesaikan untuk  $n$ . Kelihatannya lebih bagus, jika kita menerapkan beberapa penye-derhanaan untuk itu.

```
>fn &= solve(equ, n) | ratsimp; $&fn
```

$$\left[ n = \frac{\log \left( \frac{R+i Kn}{R+i K} \right)}{\log(i+1)} \right]$$

---

## Penyelesaian Soal Aljabar

---

1. Sederhanakan bentuk eksponen berikut

$$\left( \frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5} \right)^{-5}$$

Penyelesaian :

$$>\$& ((24*a^{10}*b^{-8}*c^7) / (12*a^6*b^{-3}*c^5))^{-5}$$

$$\frac{b^{25}}{32 a^{20} c^{10}}$$

2. Sederhanakan bentuk eksponen berikut

$$\left( \frac{125p^{12}q^{-14}r^{22}}{25p^8q^6r^{-15}} \right)^{-4}$$

Penyelesaian :

$$>\$& ((125*p^{12}*q^{-14}*r^{22}) / (25*p^8*q^6*r^{-15}))^{-4}$$

$$\frac{q^{80}}{625 p^{16} r^{148}}$$

3. Hitunglah

$$\frac{4(8-6)^2 - 4 \cdot 3 + 2 \cdot 8}{3^1 + 19^0}$$

Penyelesaian :

$$>& (4 * (8-6)^2 - 4 * 3 + 2 * 8) / (3^1 + 19^0)$$

5

4. Hitunglah

$$\frac{[4(8-6)^2 + 4](3 - 2 \cdot 8)}{2^2(2^3 + 5)}$$

Penyelesaian :

$$>& ((4 * (8-6)^2 + 4) * (3 - 2 * 8)) / (2^2 * (2^3 + 5))$$

- 5

5. Hitunglah

$$(3x^2 - 2x - x^3 + 2) - (5x^2 - 8x - x^3 + 4)$$

Penyelesaian :

```
> $& (3*x^2 - 2*x - x^3 + 2) - (5*x^2 - 8*x - x^3 + 4)
```

$$-2x^2 + 6x - 2$$

6. Menyelesaikan pemfaktoran

$$t^2 + 8t + 15$$

Penyelesaian :

```
> $& factor(t^2 + 8*t + 15)
```

$$(t + 3)(t + 5)$$

7. Menyelesaikan operasi yang ditunjukkan

$$(2x + 3y + z - 7) + (4x - 2y - z + 8) + (-3x + y - 2z - 4)$$

Penyelesaian :

```
> $& (2*x + 3*y + z - 7) + (4*x - 2*y - z + 8) + (-3*x + y - 2*z - 4)
```

$$-2z + 2y + 3x - 3$$

8. Hitunglah :

$$x^3 - 4x^2 + 5x - 20$$

Penyelesaian :

```
> $& factor(x^3 - 4*x^2 + 5*x - 20)
```

$$(x - 4)(x^2 + 5)$$

9. Faktorkan

$$250z^4 - 2z$$

Penyelesaian :

```
>\$&factor(250*z^4-2*z)
```

$$2 z (5 z - 1) (25 z^2 + 5 z + 1)$$

10. Faktorkan :

$$18a^2b - 15ab^2$$

Penyelesaian :

```
>\$&factor(18*a^2*b-15*a*b^2)
```

$$-3 a b (5 b - 6 a)$$

11. Faktorkan

$$25ab^4 - 25az^4$$

Penyelesaian :

```
>\$&factor(25*a*b^4-25*a*z^4)
```

$$-25 a (z - b) (z + b) (z^2 + b^2)$$

12. Faktorkan :

$$16a^7b + 54ab^7$$

Penyelesaian :

```
>\$&factor(16*a^7*b+54*a*b^7)
```

$$2 a b (3 b^2 + 2 a^2) (9 b^4 - 6 a^2 b^2 + 4 a^4)$$

13. Hitunglah :

$$m^6 + 8m^3 - 20$$

Penyelesaian :

```
>\$&factor(m^6+8*m^3-20)
```

$$(m^3 - 2) (m^3 + 10)$$

14. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(a^n + b^n)(a^n - b^n)$$

Penyelesaian :

```
> $&expand ((a^n+b^n)*(a^n-b^n))
```

$$a^{2n} - b^{2n}$$

15. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(x^{3m} - t^{5n})^2$$

Penyelesaian :

```
> $&expand ((x^(3*m)-t^(5*n))^2)
```

$$x^{6m} - 2t^{5n}x^{3m} + t^{10n}$$

16. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(a + b + c)^2$$

Penyelesaian :

```
> $&expand ((a+b+c)^2)
```

$$c^2 + 2bc + 2ac + b^2 + 2ab + a^2$$

17. Menyelesaikan operasi yang ditunjukkan

$$(2x^2 + 12xy - 11) + (6x^2 - 2x + 4) + (-x^2 - y - 2)$$

Penyelesaian :

```
> $& (2*x^2+12*x*y-11)+(6*x^2-2*x+4)+(-x^2-y-2)
```

$$12xy - y + 7x^2 - 2x - 9$$

18. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$25y^{2m} - (x^{2n} - 2x^n + 1)$$

Penyelesaiann :

```
> $&factor(25*y^(2*m)-(x^(2*n)-2*x^n+1))
```

$$(5y^m - x^n + 1) (5y^m + x^n - 1)$$

19. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$bdy^2 + ady + bcy + ac$$

Penyelesaian :

```
> $&factor(b*d*y^2+a*d*y+b*c*y+a*c)
```

$$(b y + a) (d y + c)$$

20. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$4x^{2n} - 4x^n - 3$$

Penyelesaian :

```
> $&factor(b*d*y^2+a*d*y+b*c*y+a*c)
```

$$(b y + a) (d y + c)$$

---

---

## BAB 3

---

# KB PEKAN 5-6: MENGGUNAKAN EMT UNTUK MENGAMBAR GRAFIK 2 DIMENSI (2D)

### Menggambar Grafik 2D dengan EMT

---

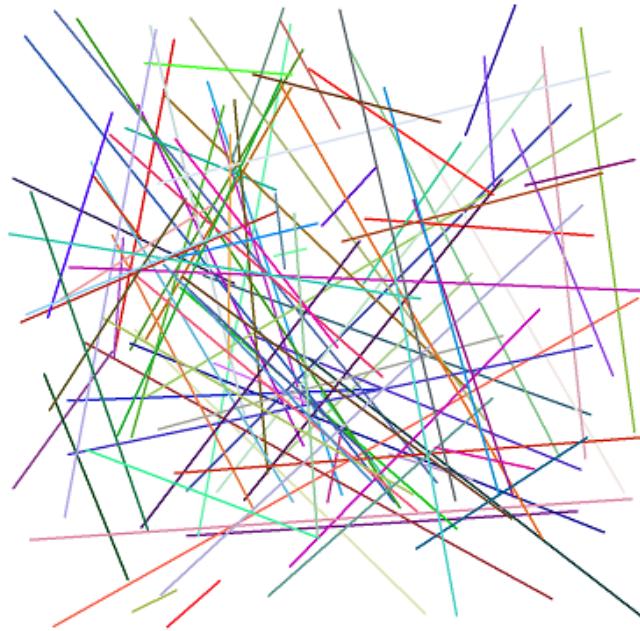
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D). **Plot Dasar**

---

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
```



```
>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.

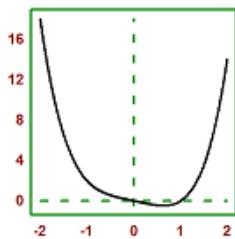
Untuk contoh lain, kami menggambar plot sebagai sisipan di plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan 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);
```

```
Function barclear not found.
Try list ... to find functions!
```

```
Error in :
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.

**As-**

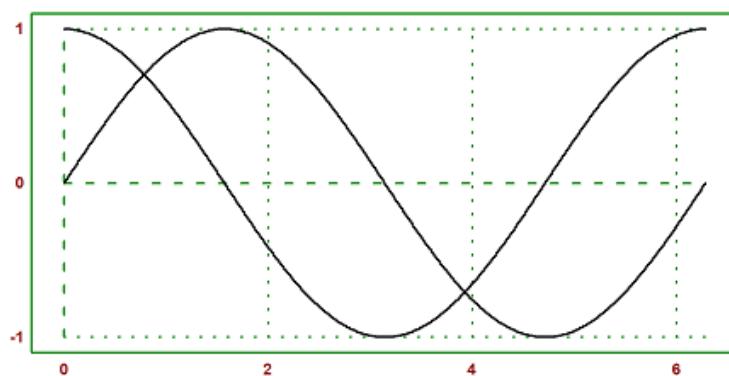
## pek Plot

---

Plot default menggunakan jendela plot persegi. Anda dapat mengubah ini dengan fungsi aspek(). 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 default plot termasuk rasio aspek.

## 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 berbayangan.

## Plot

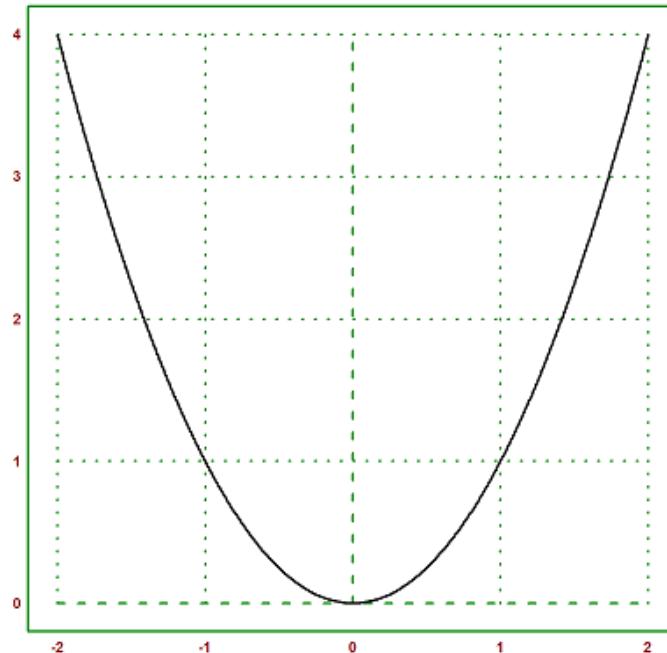
### Ekspresi atau Variabel

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

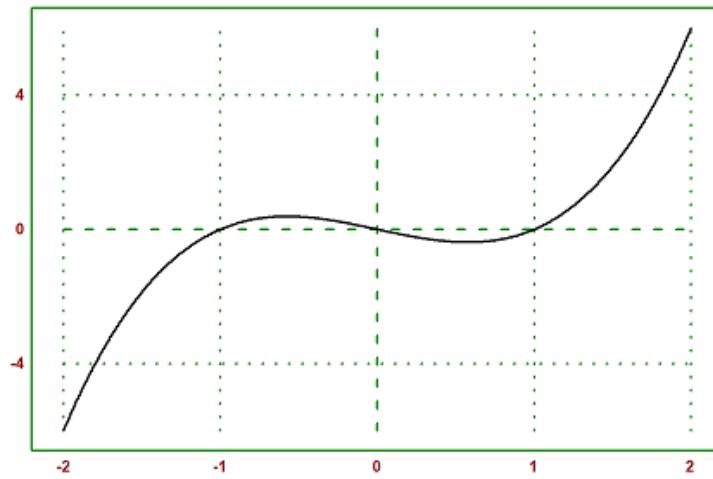
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

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.

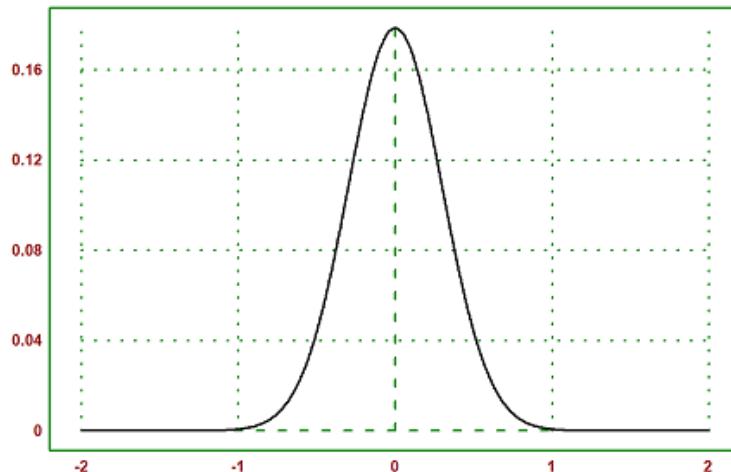
```
>plot2d("x^2"):
```



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



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

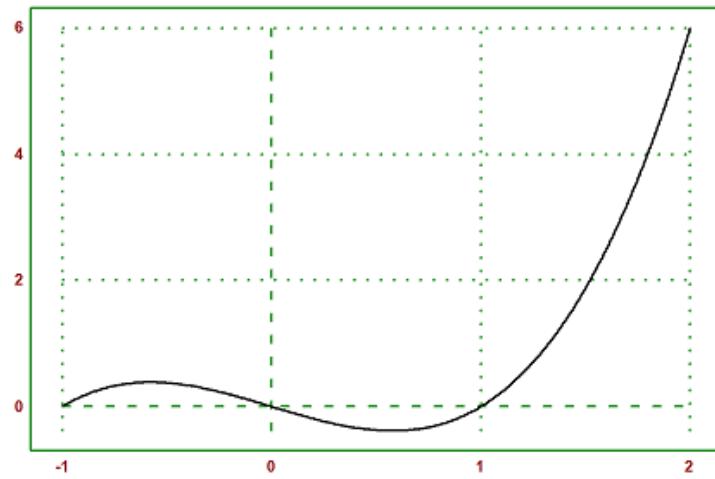


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

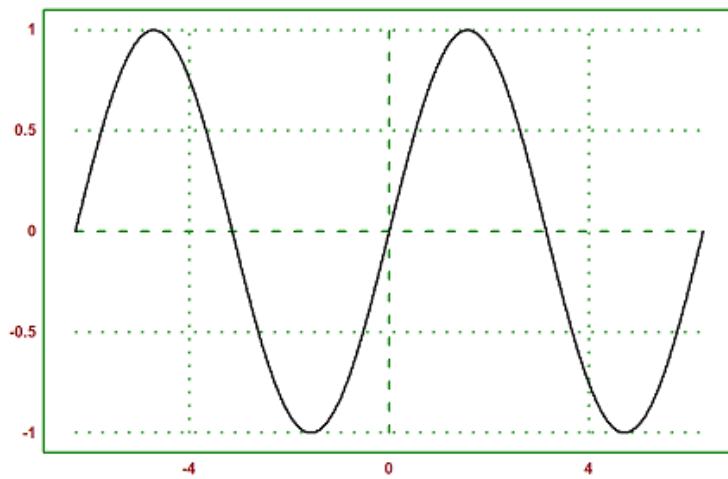
Rentang plot diatur dengan parameter yang ditetapkan sebagai berikut

- 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)

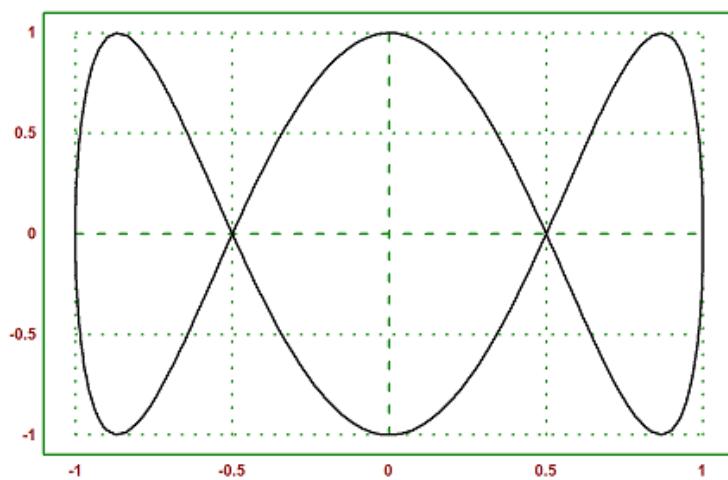
```
>plot2d("x^3-x", -1, 2):
```



```
>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

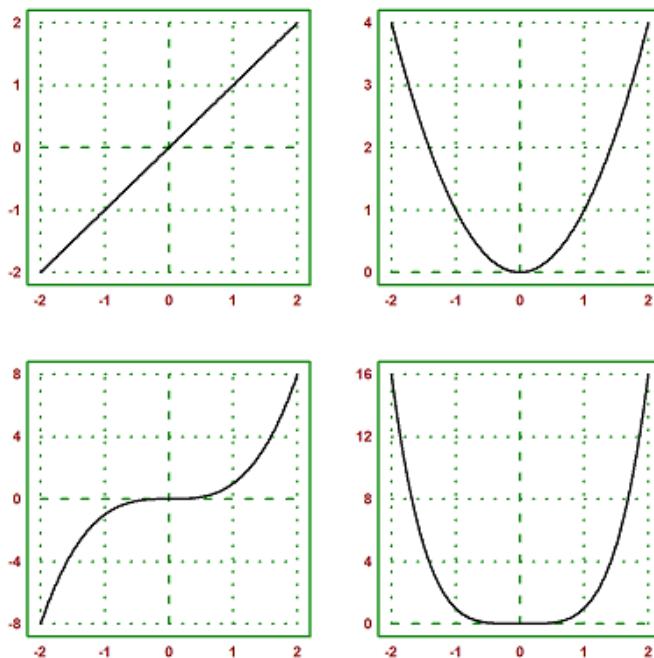
- di jendela terpisah yang dapat diubah ukurannya, - di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

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

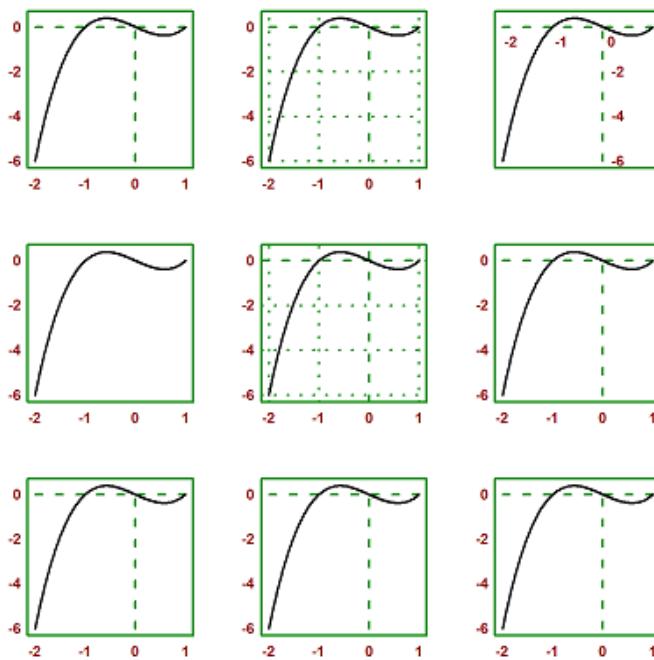
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh, kami memplot  $x^1$  hingga  $x^4$  menjadi 4 bagian jendela. `figure(0)` mengatur ulang jendela default.

```
>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):
```

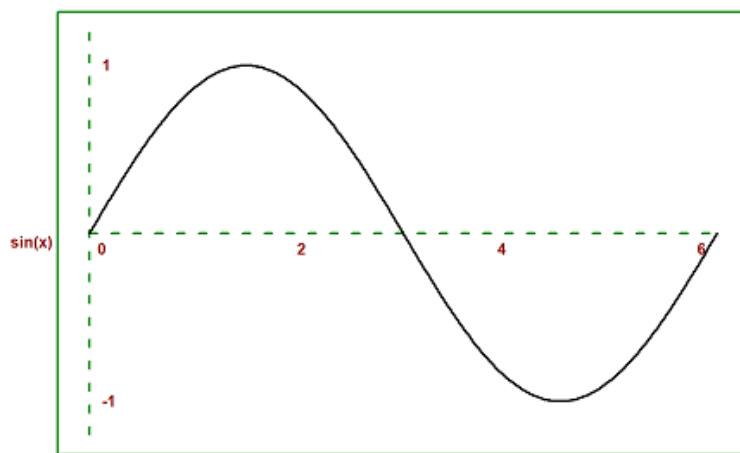


Jika argumen ke plot2d() adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

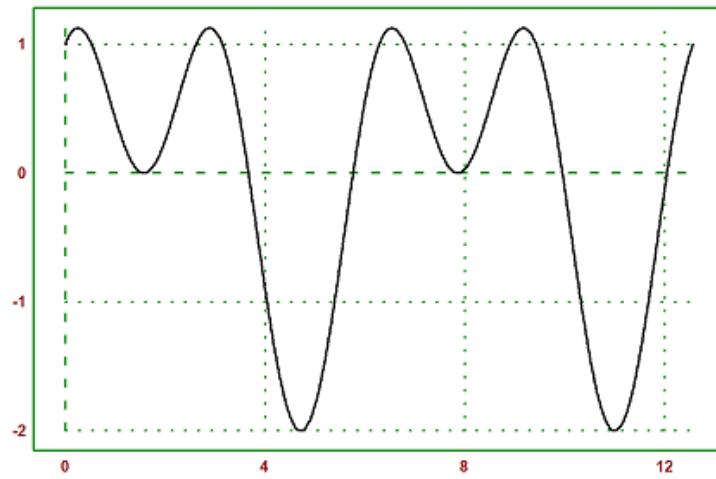
Atau, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan sebagai a=... dll.

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);
```

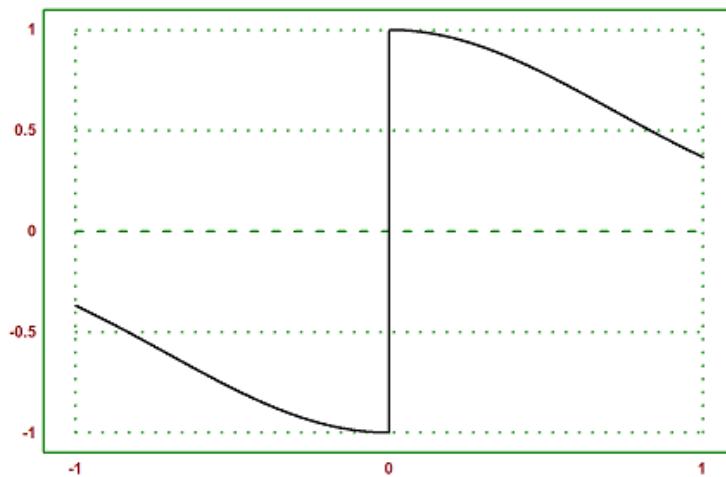


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "gambar". Mereka juga digunakan oleh ekspor HTML.

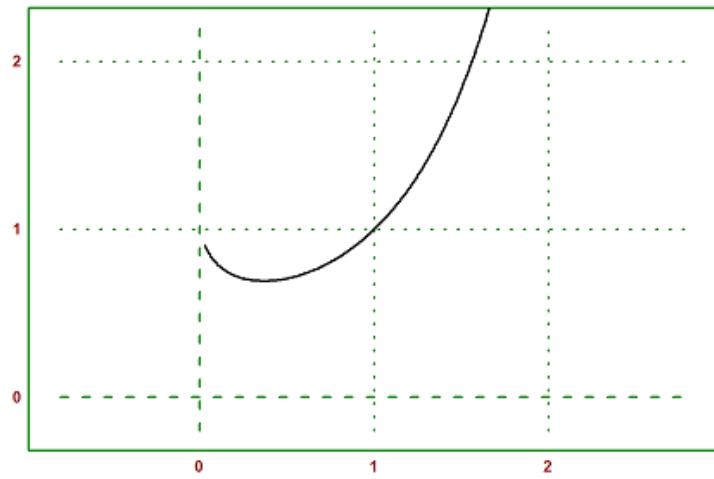
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

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):
```

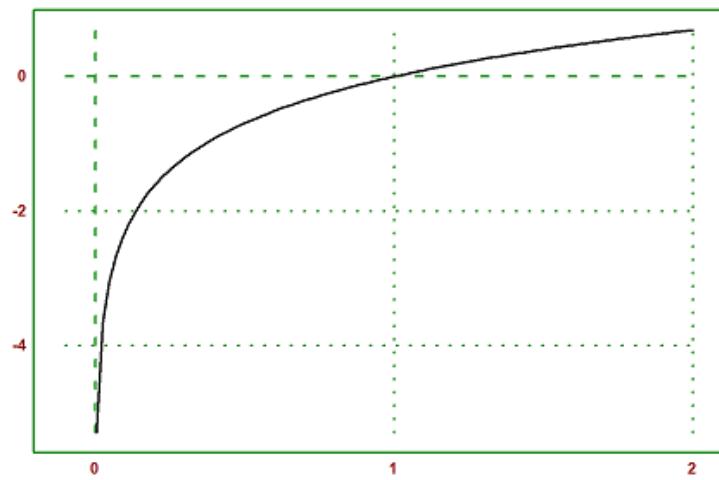


```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



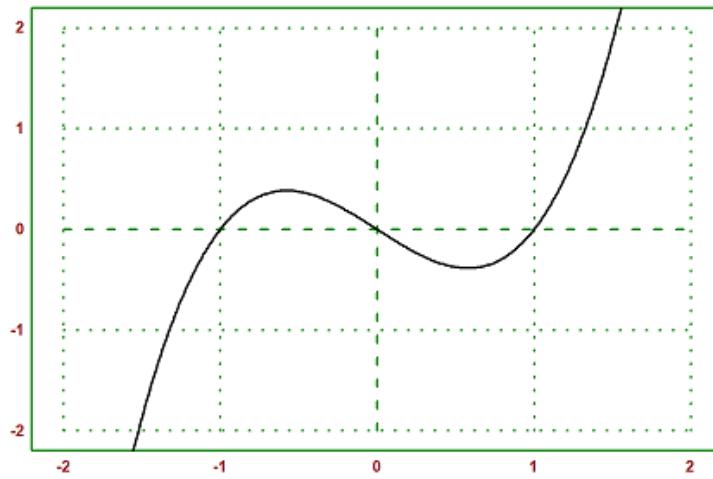
Perhatikan bahwa  $x^x$  tidak definisikan untuk  $x \leq 0$ . Fungsi plot 2d menangkap kesalahan ini, dan mulai merencanakan segera

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

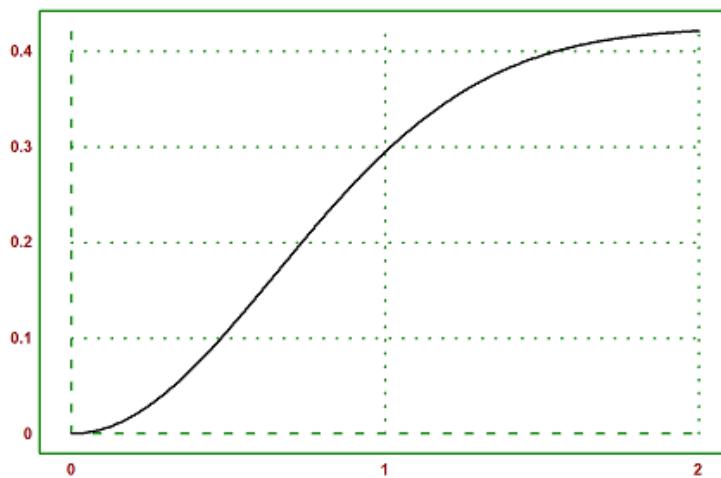


Parameter square=true (atau >square) memilih y-range secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

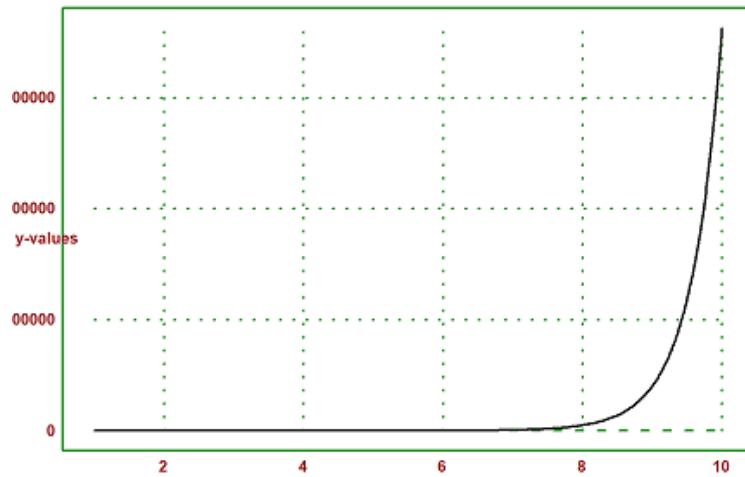


```
>plot2d('' integrate("sin(x)*exp(-x^2)", 0, x)'', 0, 2): // plot integral
```



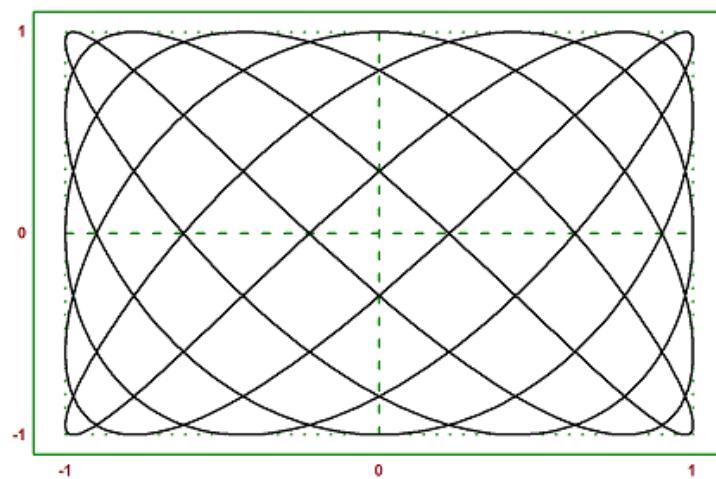
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di plot2d().

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

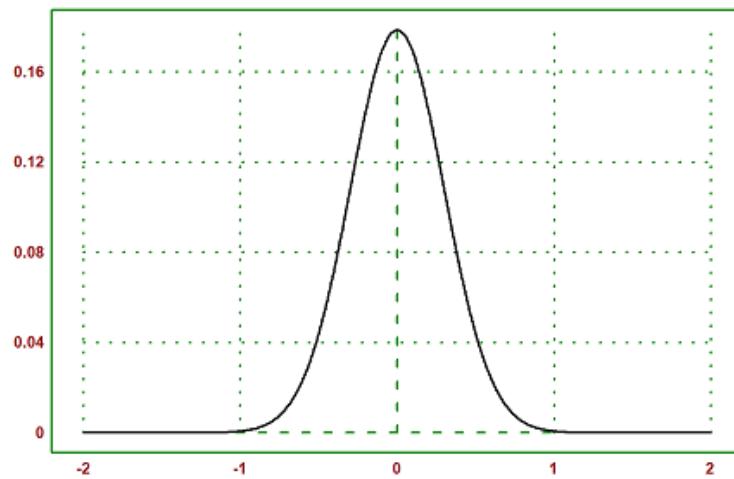


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

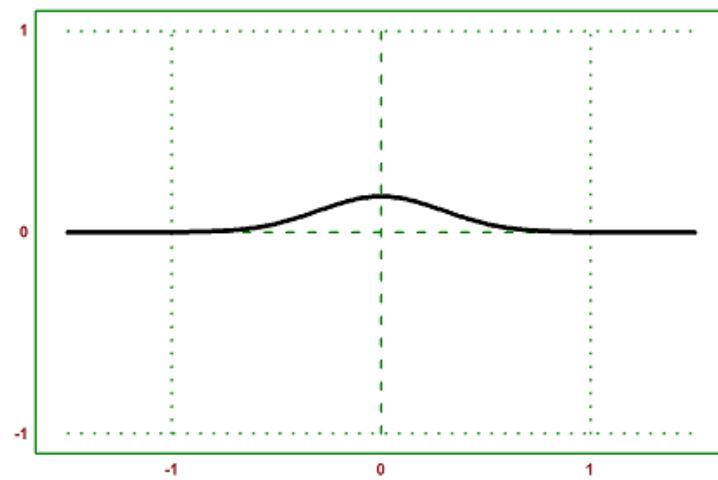
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



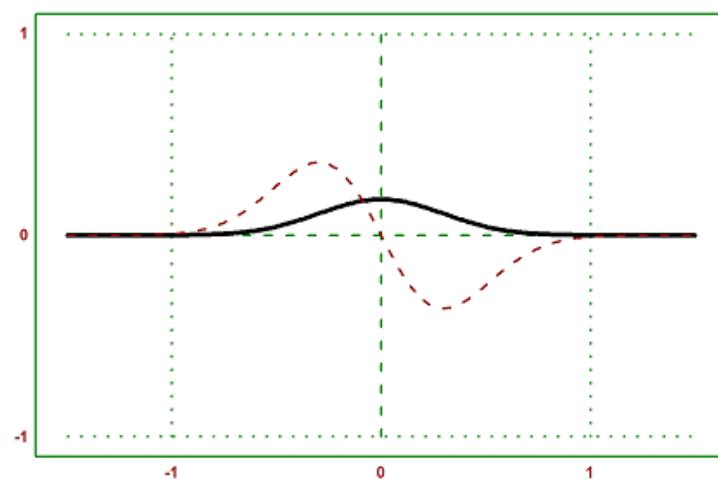
```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2); // plot from -2 to 2
```



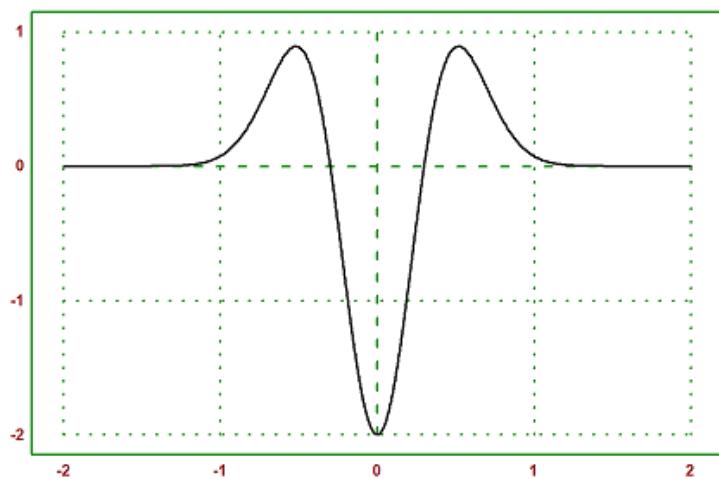
```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



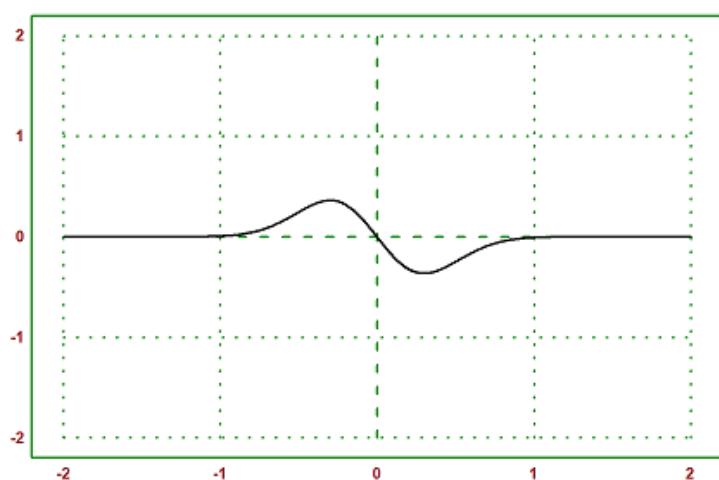
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



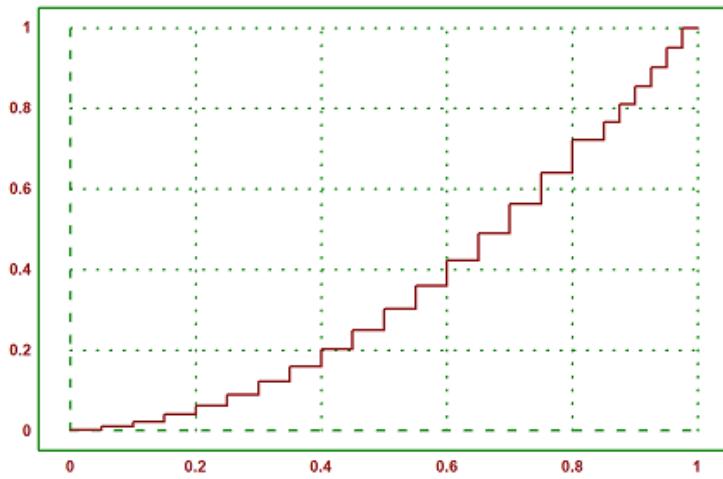
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



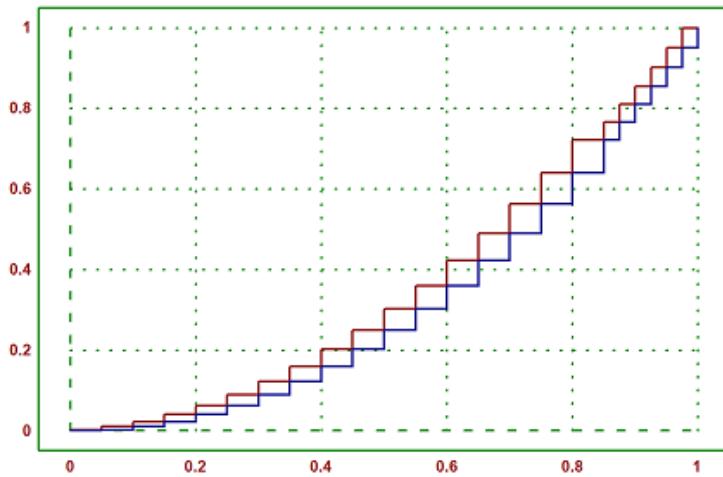
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```



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



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

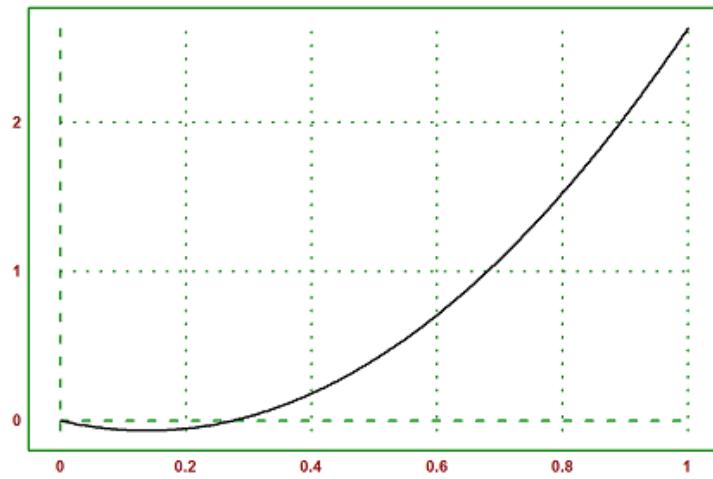


## Fungsi dalam satu Parameter

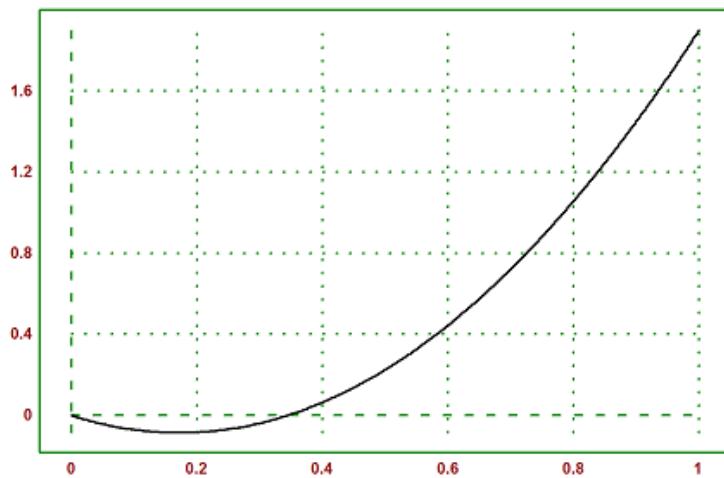
Fungsi plot yang paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.

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
```

Syntax error in expression, or unfinished expression!

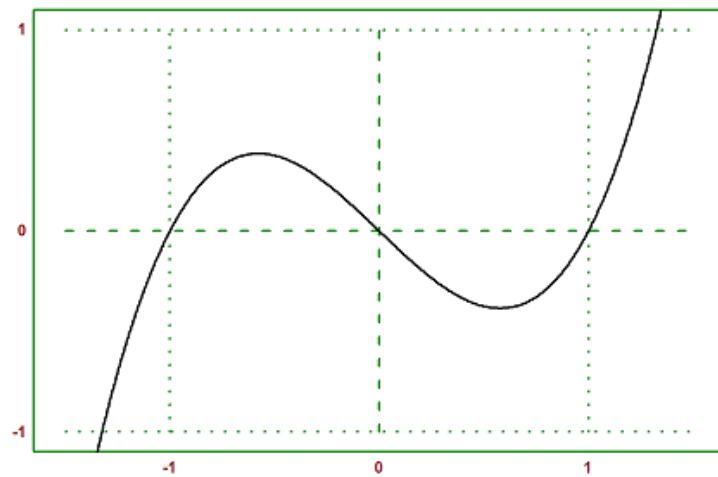
Error in :  
 plot2d({{"f",0.2}},0,1): // plot with a=0.2  
 ^

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

Syntax error in expression, or unfinished expression!

Error in :  
 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

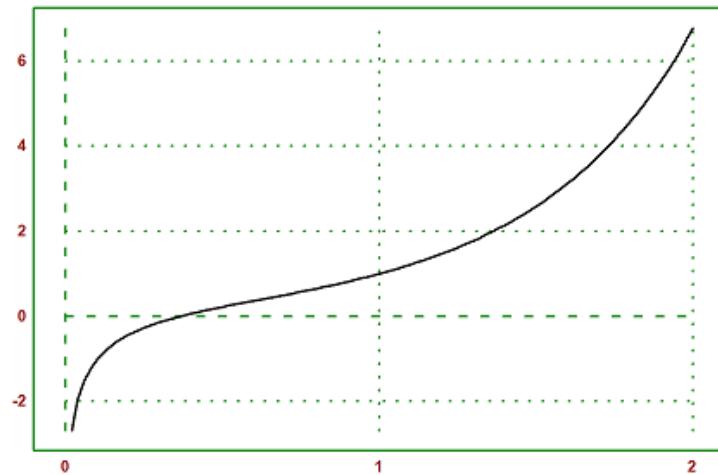
- ekspresi atau ekspresi simbolik dalam  $x$  - fungsi atau fungsi simbolis dengan nama sebagai "f" - fungsi simbolis hanya dengan nama f

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

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

$$\frac{x}{x} \cdot (\log(x) + 1)$$

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

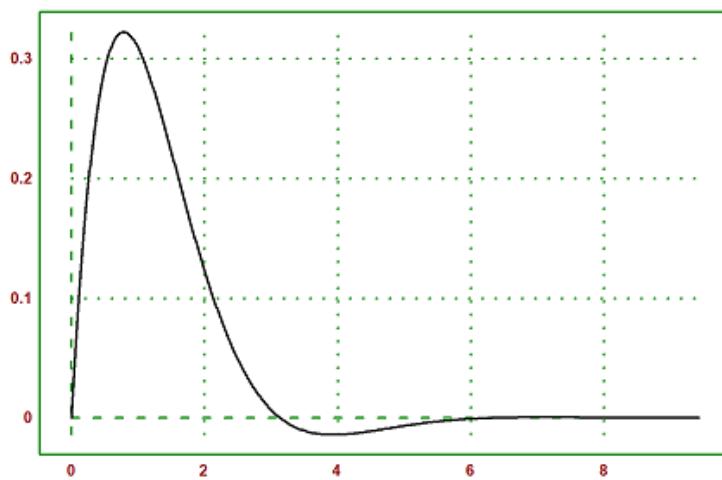


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.

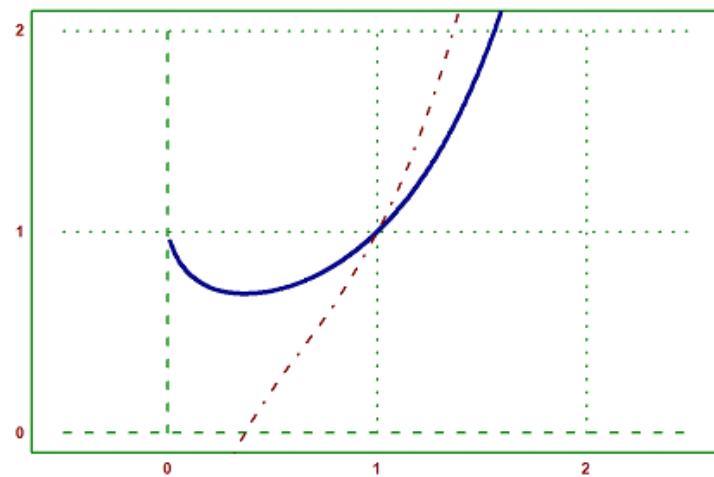
```
>expr &= sin(x)*exp(-x)
```

$$\frac{e^{-x} \sin(x)}{x}$$

```
>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="-.-"):
```



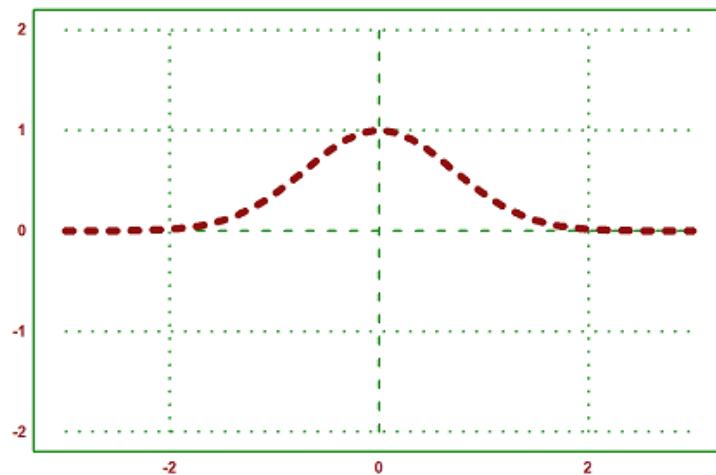
Untuk gaya garis ada berbagai pilihan.

- `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.

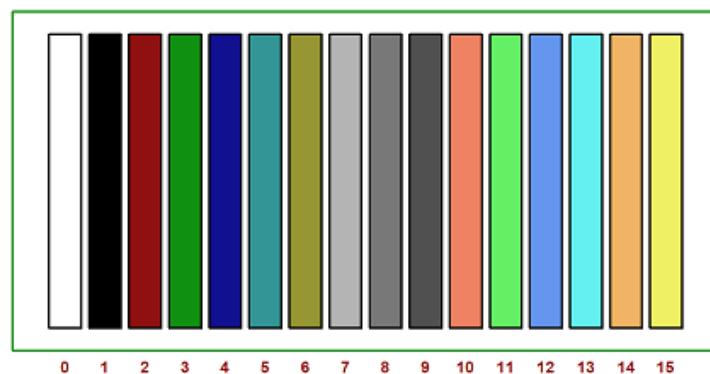
- 0.15: indeks warna default. - konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye terang, kuning - `rgb(merah, hijau, biru):` parameter adalah real dalam [0,1].

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



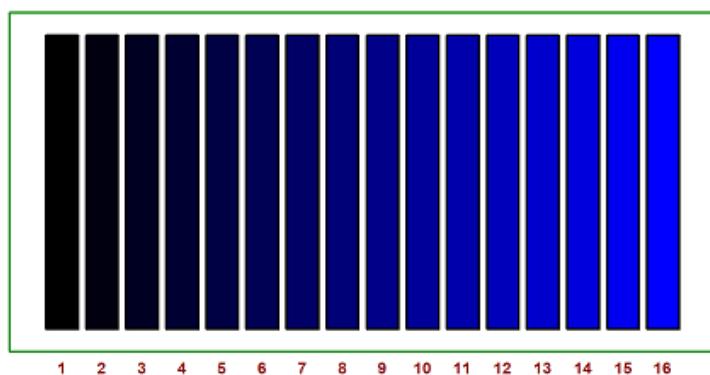
Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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



Tapi Anda bisa menggunakan warna apa saja.

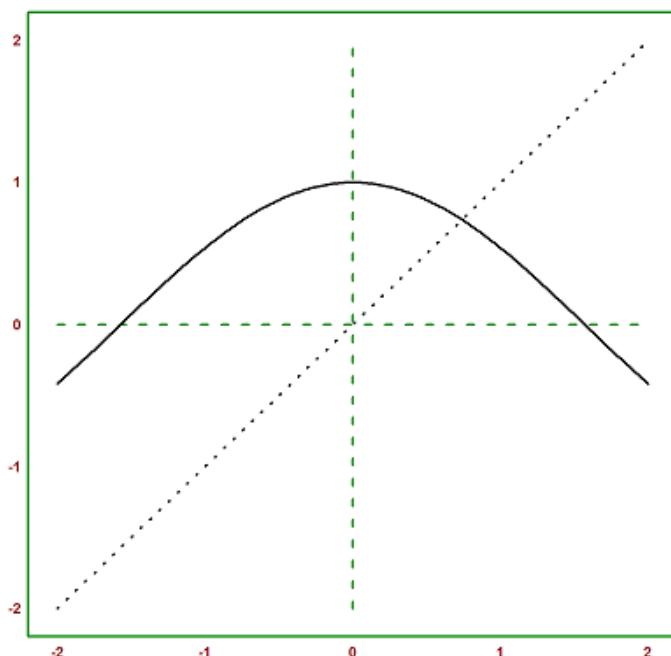
```
>columnsplot(ones(1,16), grid=0, color=rgb(0,0,linspace(0,1,15))) :
```



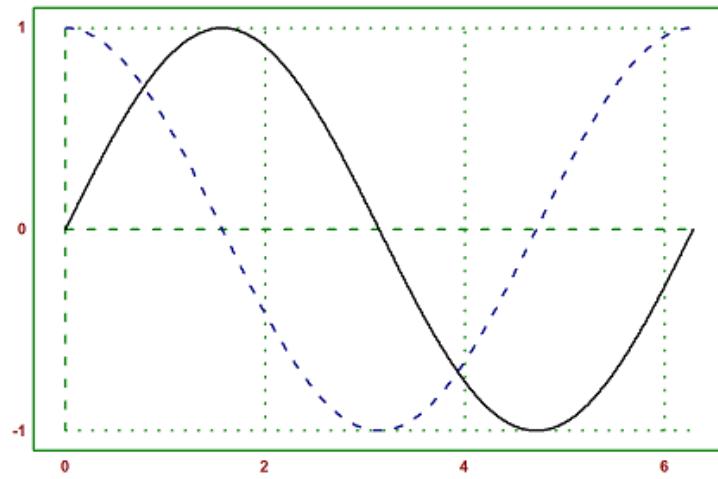
## Menggambar Beberapa Kurva pada bidang koordinat yang sama

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):
```

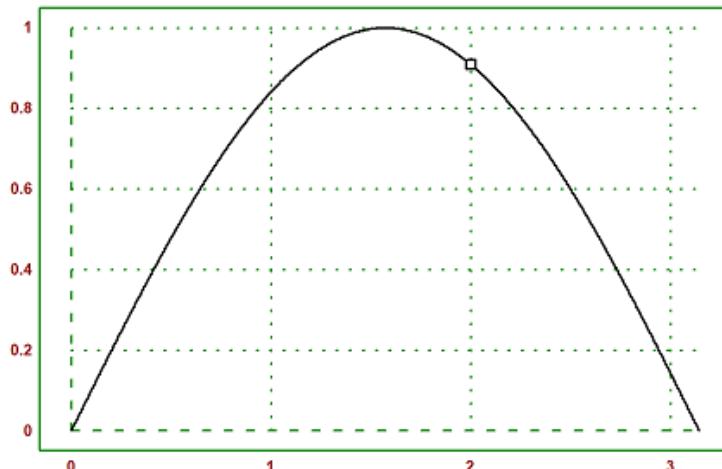


```
>aspect(1.5); plot2d("sin(x)", 0, 2pi); plot2d("cos(x)", color=blue, style="--", >add):
```



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

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



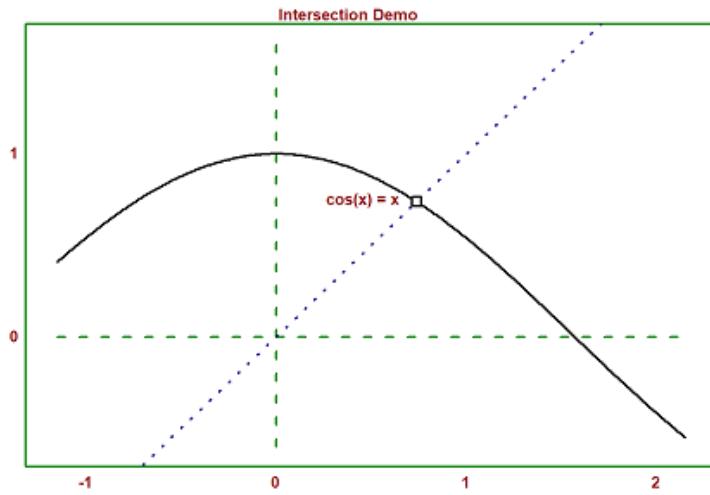
Kami menambahkan titik persimpangan dengan label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasilnya ke dalam notebook. Kami juga menambahkan judul ke plot.

```
>plot2d(["cos(x)", "x"], r=1.1, cx=0.5, cy=0.5, ...
> color=[black,blue], style=[ "-", "." ], ...
> grid=1);
```

Illegal parameter after named parameter!

```
Error in :
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):
```



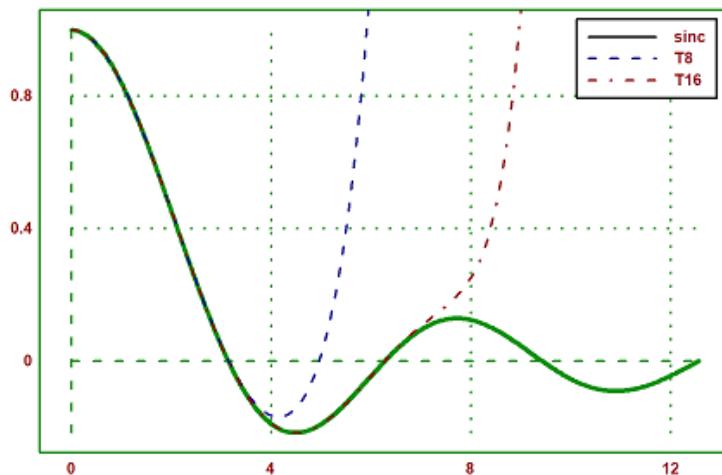
Dalam demo berikut, kami memplot fungsi  $\text{sinc}(x)=\sin(x)/x$  dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolis. Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Yang kedua dan yang ketiga memiliki set flag `>add`, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsi.

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

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

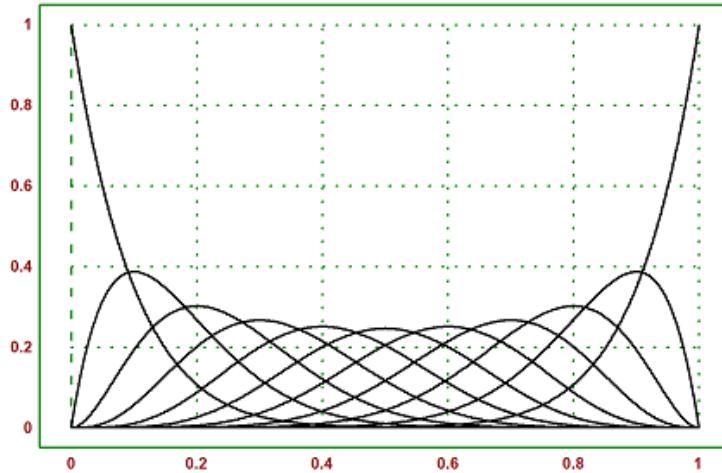
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
> plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
> plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-."); ...
> labelbox(["sinc","T8","T16"],styles=["-","--","-."], ...
> colors=[black,blue,red]):
```



Dalam contoh berikut, kami menghasilkan Bernstein-Polinomial.

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

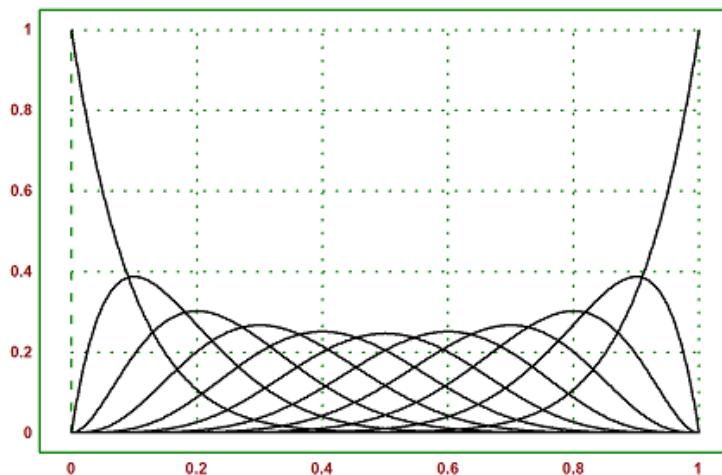
```
>plot2d("(1-x)^10",0,1); // plot first function  
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;  
>insimg;
```



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);
```



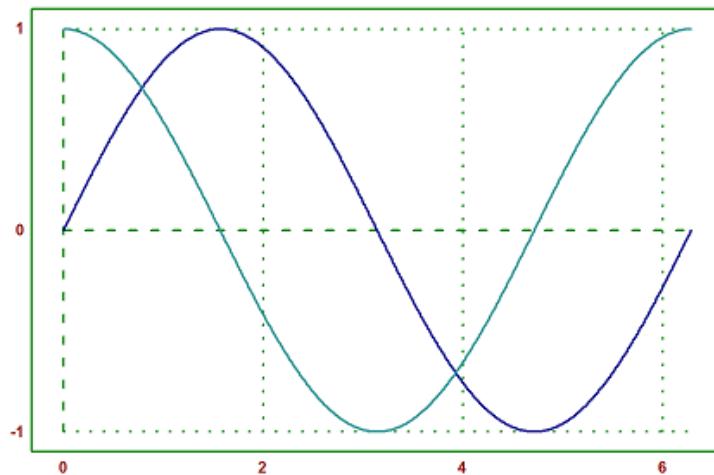
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

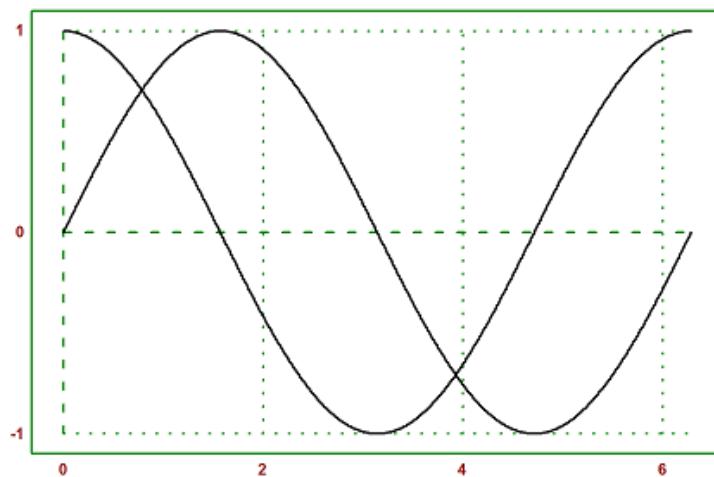
```
Argument for color must be real!
plot2d:           ccc=_color(color); l=_linewidth(thickness);
```

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().

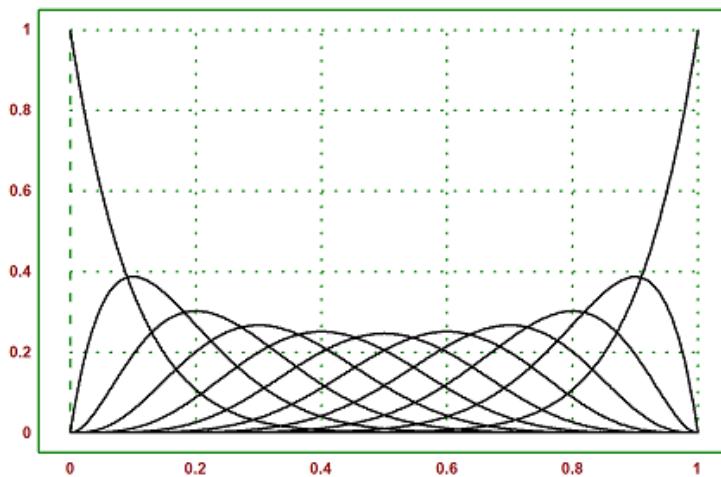
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

```
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):
```

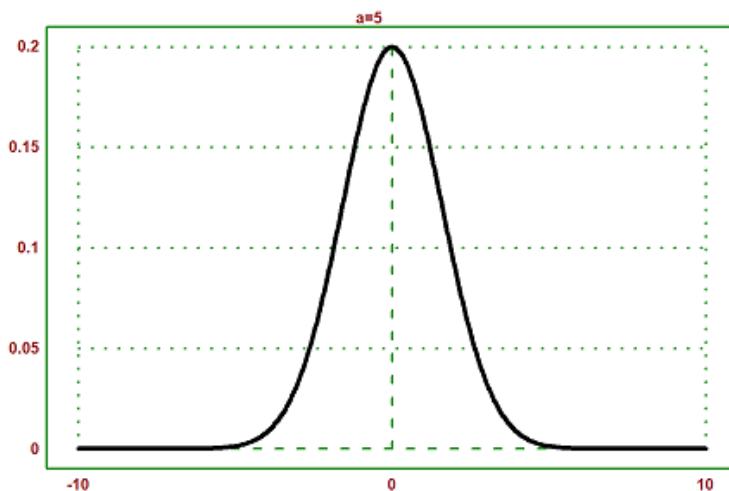
```
Argument for color must be real!
plot2d:           ccc=_color(color); l=_linewidth(thickness);
```

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); ...
```

```
Syntax error in expression, or unfinished expression!
```

```
Error in :
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)):
```

Function getspectral not found.  
Try list ... to find functions!

Error in :  
x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10)):  
^

## Label Teks

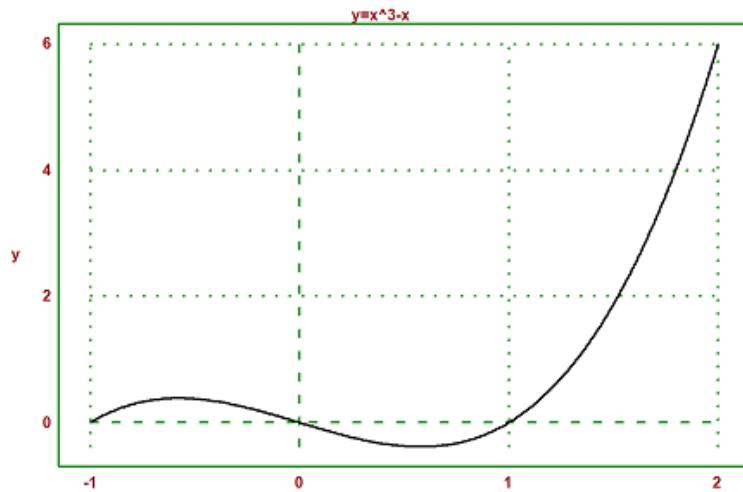
---

Dekorasi sederhana bisa

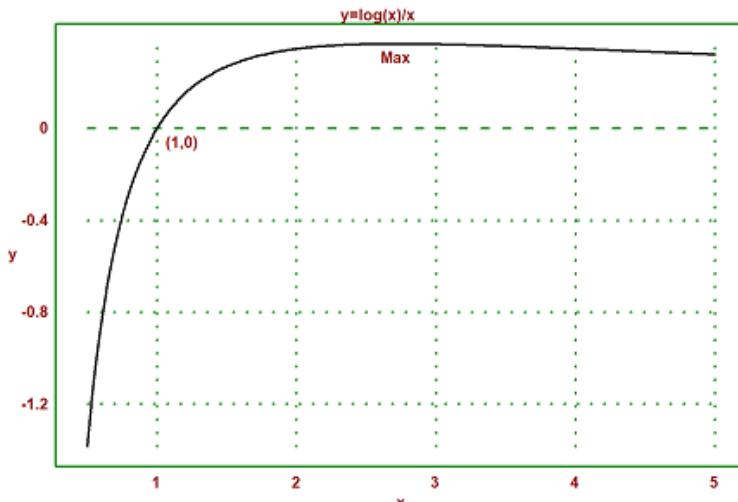
- judul dengan judul="..." - x- dan y-label dengan xl="...", yl="..." - label teks lain dengan label("...",x,y)

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"):
```

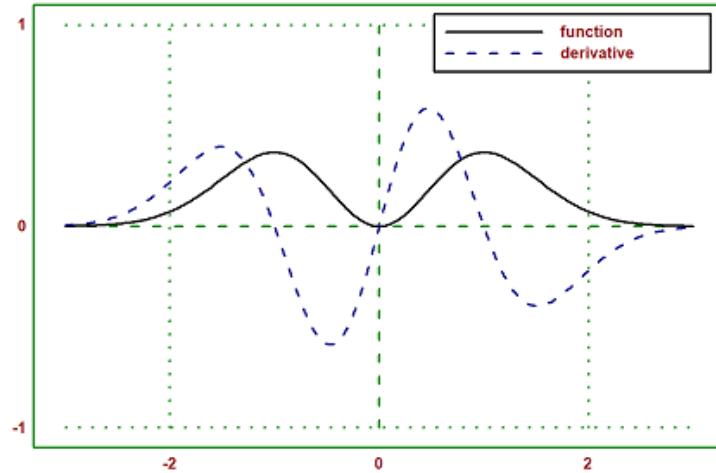


```
>expr := "log(x)/x"; ...
> plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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):
```

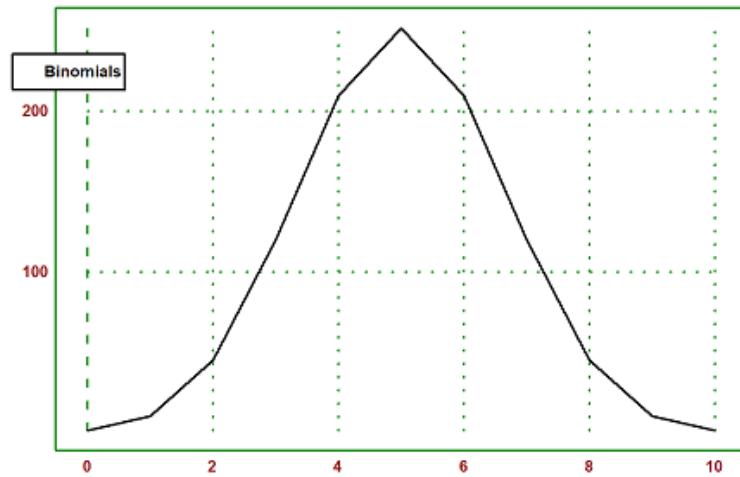


Kotak ditambatkan di kanan atas secara default, tetapi > kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebaranya otomatis.

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

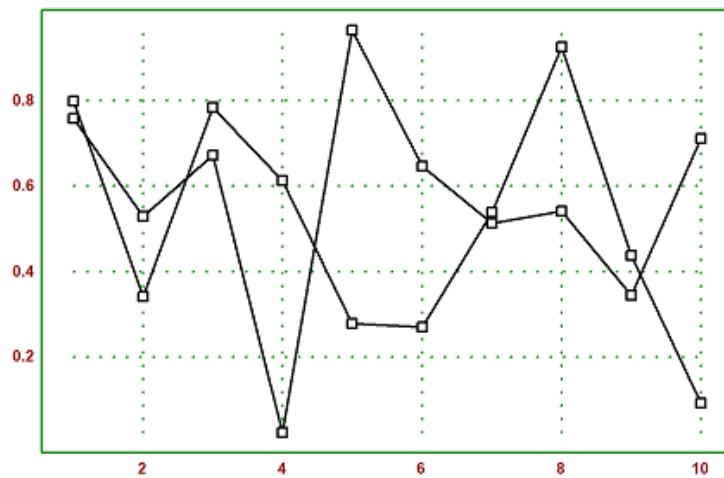
Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

```
>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):
```

Function latex not found.

Try list ... to find functions!

Error in :

```
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 &rarr; x3 - x"):
```

Variable u not found!

Error in :  
plot2d("x^3-x",title=u"x &rarr; x<sup>3</sup> - x"):  
          ^

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

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

Variable u not found!

Error in :  
plot2d("sinc(x)",0,2pi,xl="x",yl=u"x &rarr; sinc(x)",>vertical):  
          ^

## LaTeX

---

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "lateks" dan "dvipng" 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"), ...
```

Function latex not found.  
Try list ... to find functions!

Error in :  
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):
```

```
Function xtick not found.
Try list ... to find functions!
```

```
Error in :
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)); ...
```

```
Function latex not found.
Try list ... to find functions!
```

```
Error in :
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 \leq 0 \end{cases}"), ...
>  x=0.7,y=0.2):
```

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.

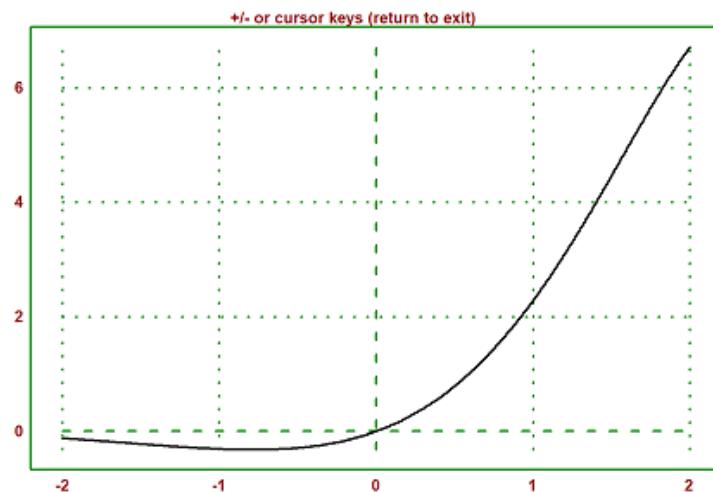
```
>plot2d({{"x^3-a*x"},a=1},>user,title="Press any key!":
```

Syntax error in expression, or unfinished expression!

Error in :  

```
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) ...
```

```

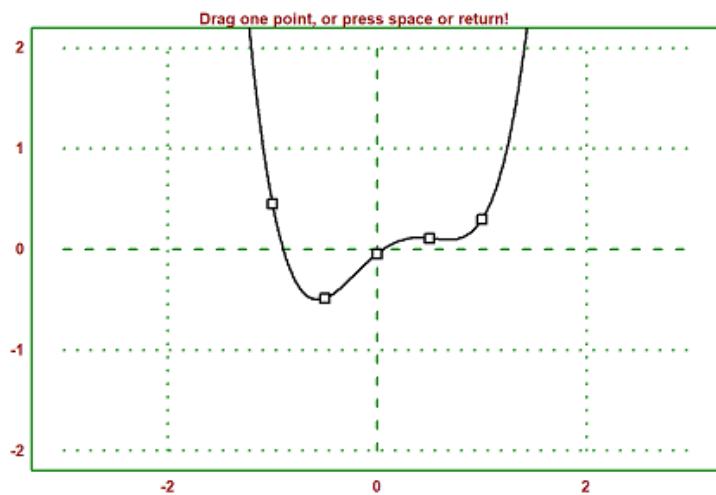
d=interp(xp,yp);
plot2d("interpval(xp,d,x)";d,xp,r=2);
plot2d(xp,yp,>points,>add);
if select>0 then
  plot2d(xp[select],yp[select],color=red,>points,>add);
endif;
title("Drag one point, or press space or return!");
endfunction

```

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):
```



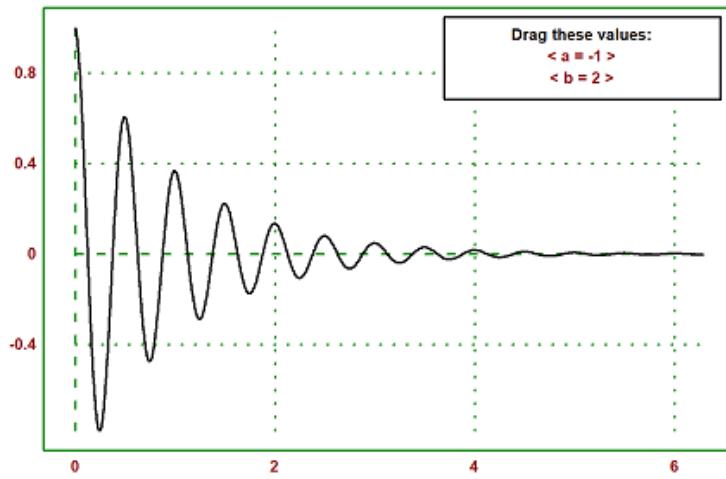
Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

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):
```

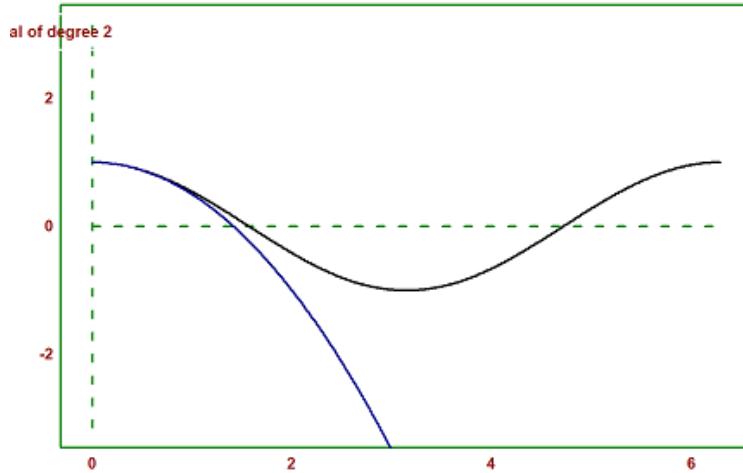


Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>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.

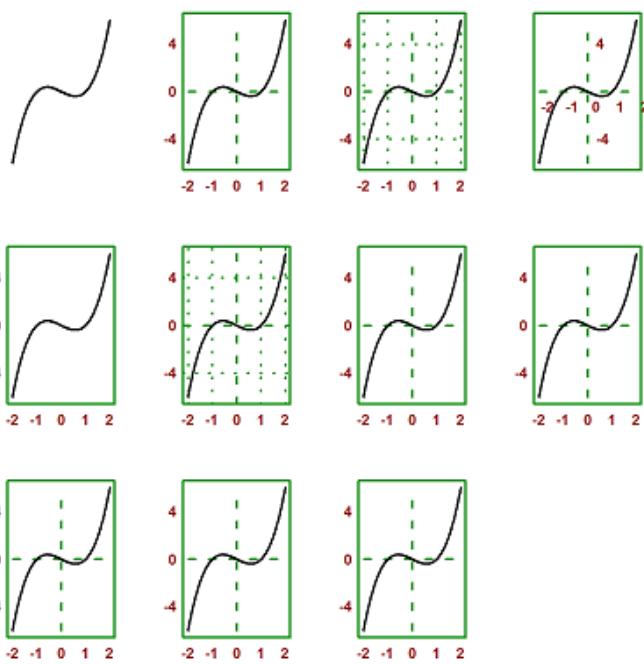
```
>function dragtest ...  
  
plot2d(none,r=1,title="Drag with the mouse, or press any key!");  
start=0;  
repeat  
{flag,m,time}=mousedrag();  
if flag==0 then return; endif;  
if flag==2 then  
    hold on; mark(m[1],m[2]); hold off;  
    endif;  
end  
endfunction  
  
>dragtest // lihat hasilnya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

## Gaya Plot 2D

---

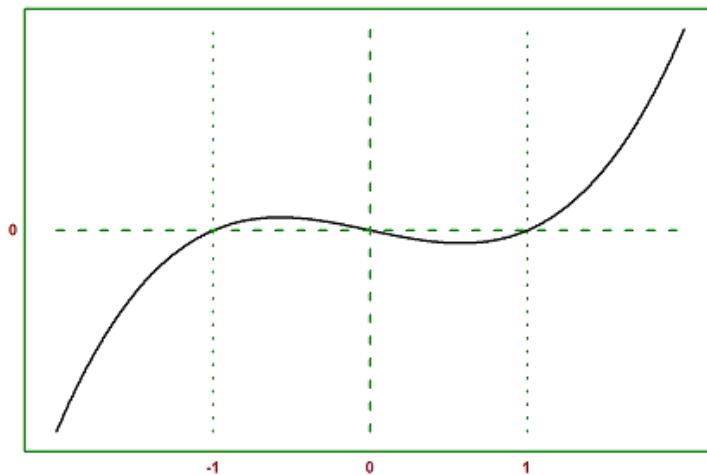
Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

```
>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
```

```
Function latex not found.  
Try list ... to find functions!  
  
Error in :  
title(latex("y=e^x")); // title above the plot  
^
```

```
>xlabel(latex("x")); // "x" for x-axis
```

```
Function latex not found.  
Try list ... to find functions!
```

```
Error in :  
xlabel(latex("x")); // "x" for x-axis  
^
```

```
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
```

```
Function latex not found.  
Try list ... to find functions!
```

```
Error in :  
ylabel(latex("y"),>vertical); // vertical "y" for y-axis  
^
```

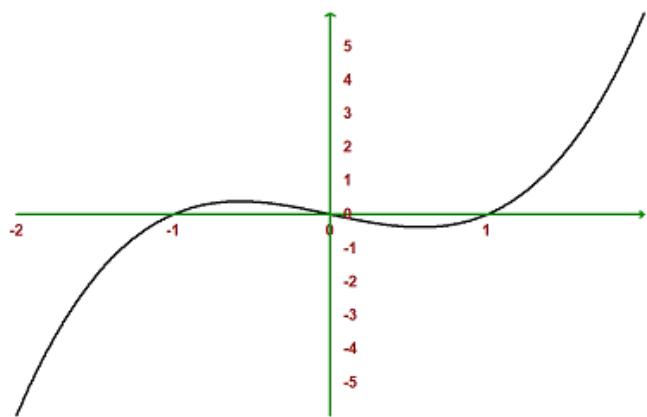
```
>label(latex("(0,1)'),0,1,color=blue): // label a point
```

```
Function latex not found.  
Try list ... to find functions!
```

```
Error in :  
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="->"):
```

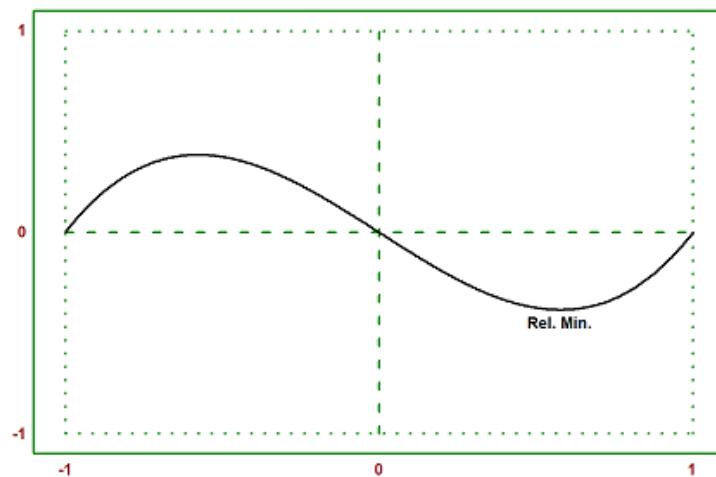


Teks pada plot dapat diatur dengan `label()`. Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

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

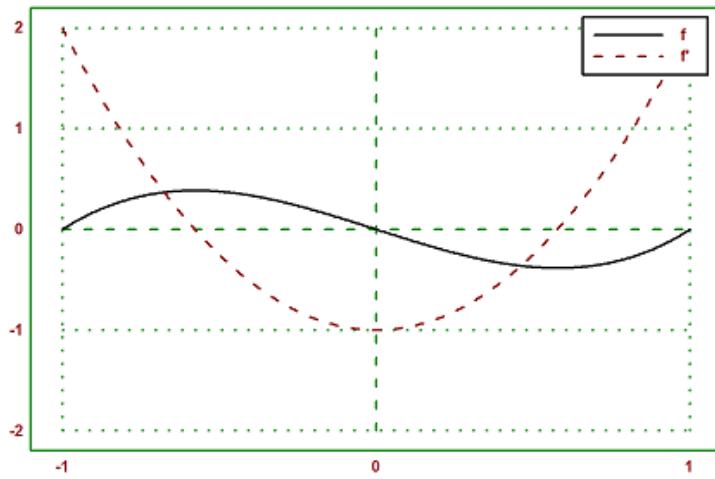
$$x^3 - x$$

```
>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
```

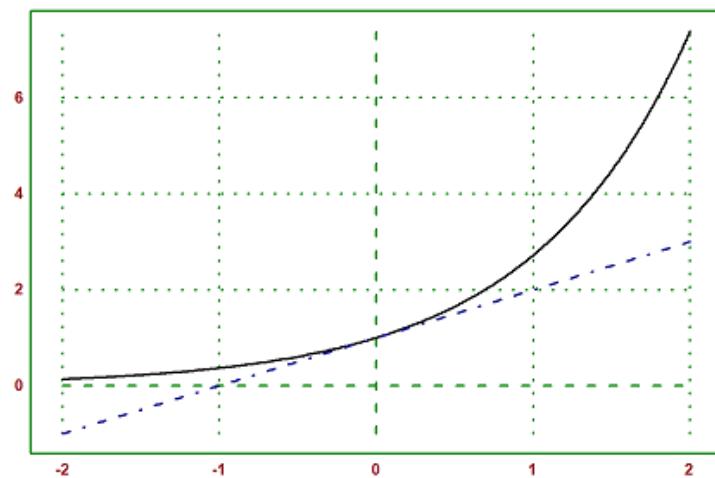


Ada juga kotak teks.

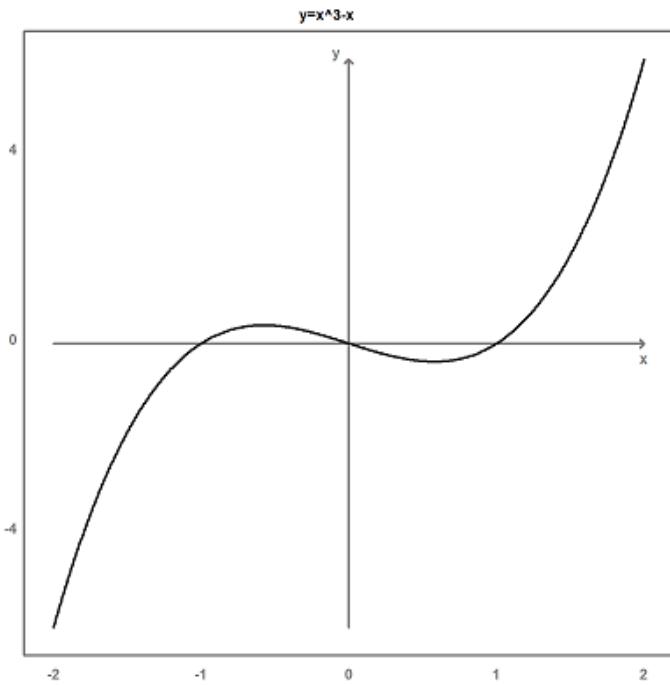
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-", "--"],[black,red]): // label box
```



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



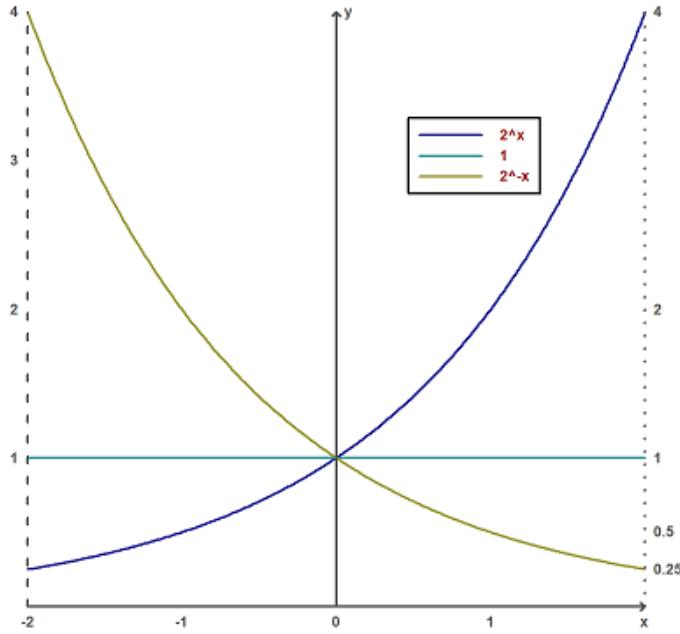
```
>gridstyle("->", color=gray, textcolor=gray, framecolor=gray); ...
> plot2d("x^3-x", grid=1); ...
> settitle("y=x^3-x", color=black); ...
> label("x", 2, 0, pos="bc", color=gray); ...
> label("y", 0, 6, pos="cl", color=gray); ...
> reset():
```



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"\u03c0;","u"2\u03c0;"],style="-",>ticks,>zero); ...

Variable u not found!

Error in :
  xaxis(-1.1,(0:2)*pi,xt=["0",u"\u03c0;","u"2\u03c0;"],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"\u03c6"); ylabel(u"f(\u03c6);"):
```

## 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.

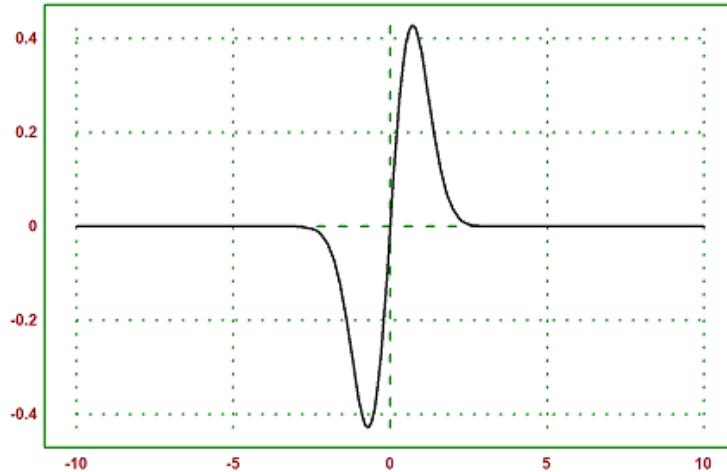
Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai  $x$ , dan satu atau beberapa baris nilai  $y$ . Dari rentang dan nilai- $x$ , fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. 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):
```

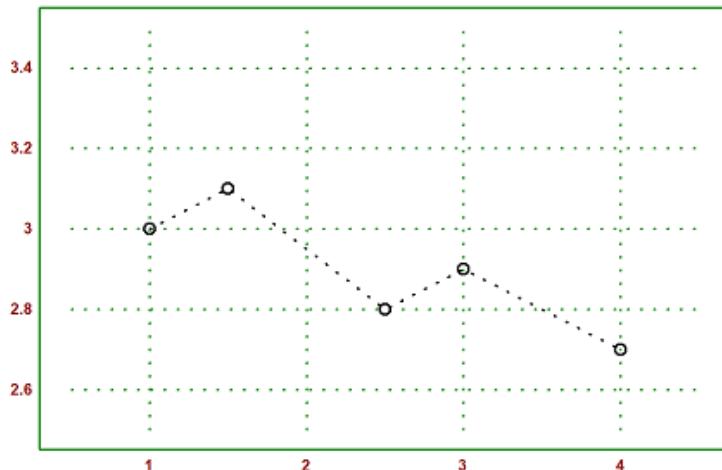


Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

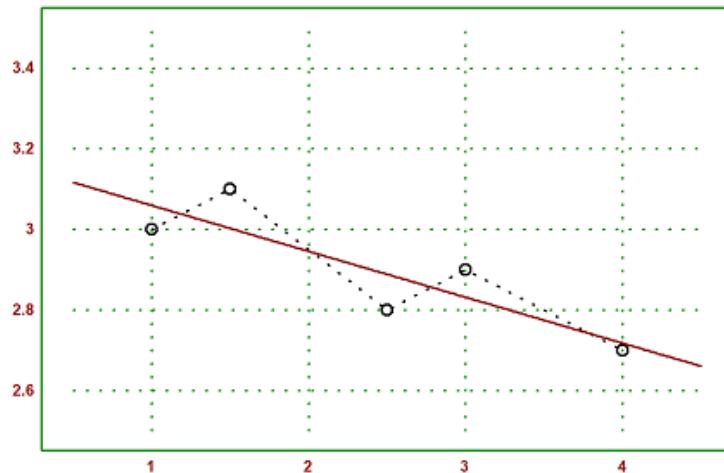
- style="...": Pilih dari "[", "<>", "o", ".", "..", "+", "\*", "[", "<>", "o", "..", "", "|".

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
```



## Menggambar Daerah Yang Dibatasi Kurva

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

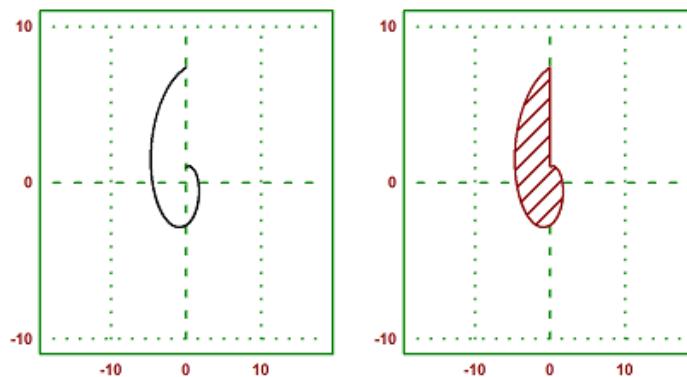
- terisi=benar mengisi plot. - style="...": Pilih dari "", "/", "\", "\\"/. - fillcolor: Lihat di atas untuk warna yang tersedia.

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)
```

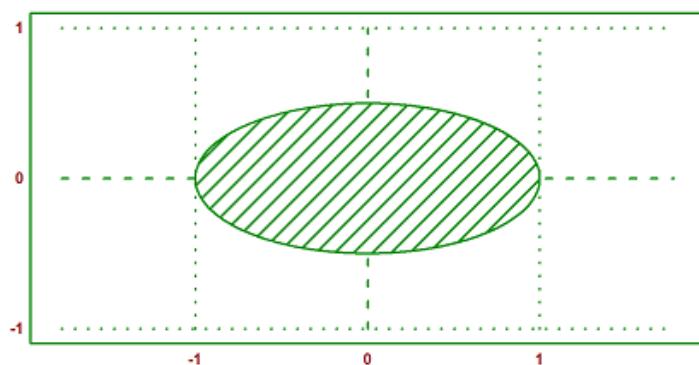
1.5

```
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```

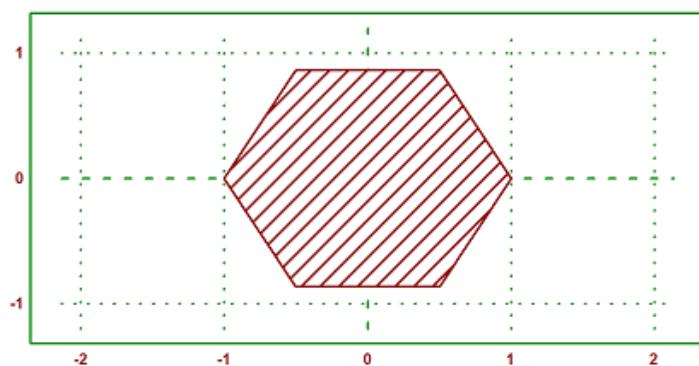


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

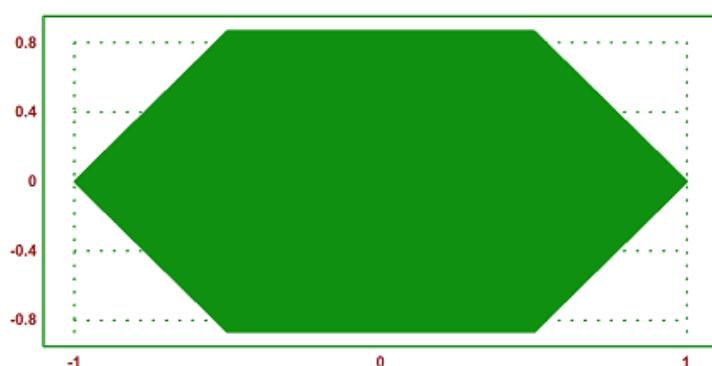
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



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

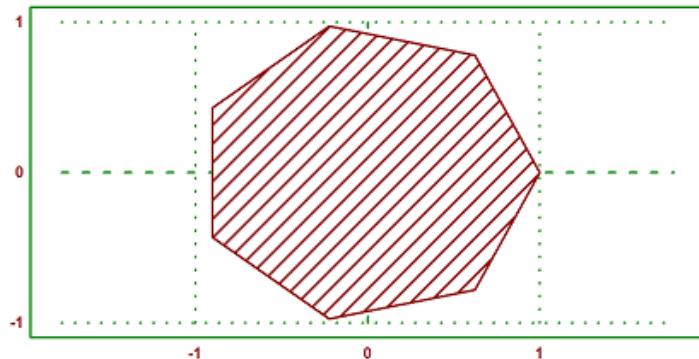


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



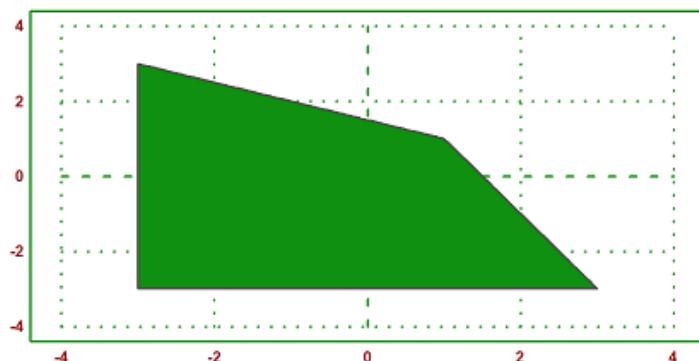
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>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):
```

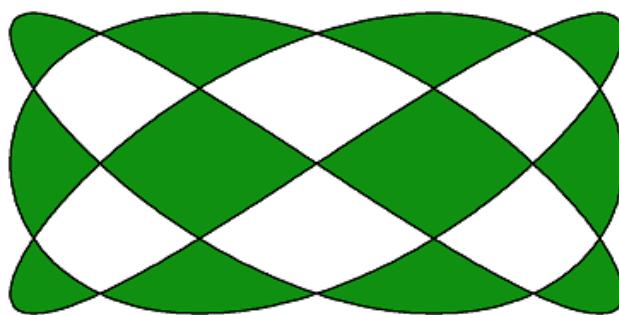


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

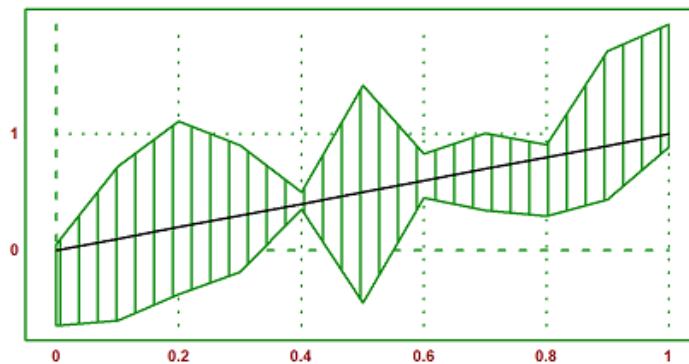
Kami sekarang memiliki vektor  $x$  dan  $y$  nilai. `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk isi.

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



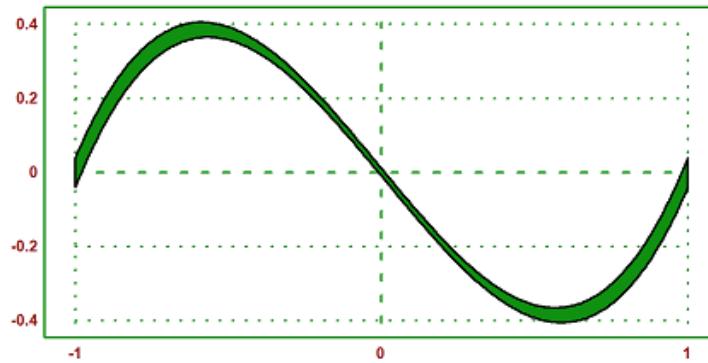
Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas. Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

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



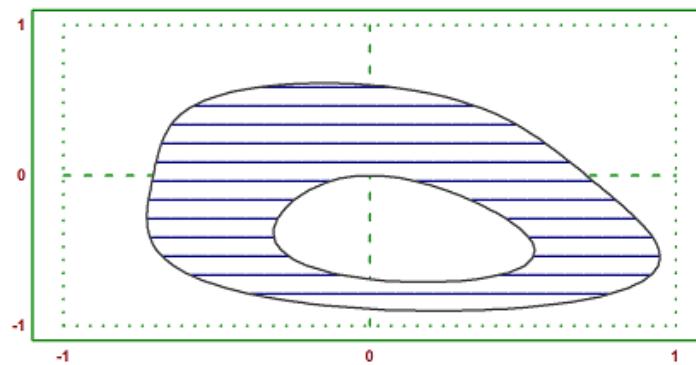
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>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 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

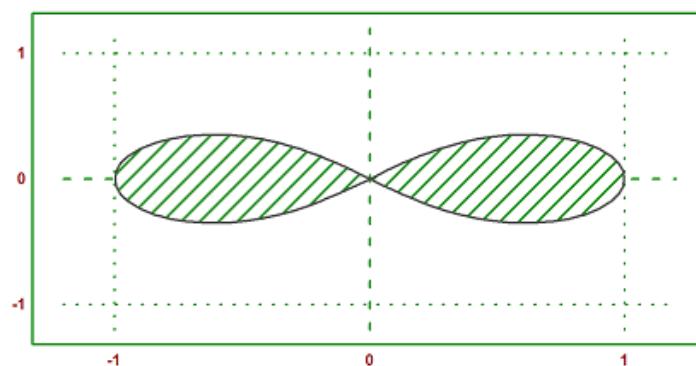
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



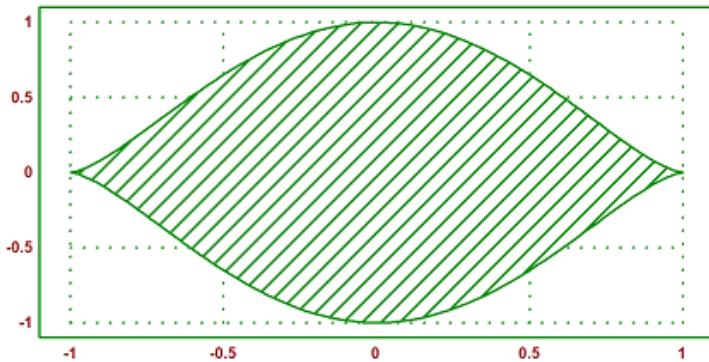
Kami juga dapat mengisi rentang nilai seperti

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

```
>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="/") :
```



## Grafik Fungsi Parametrik

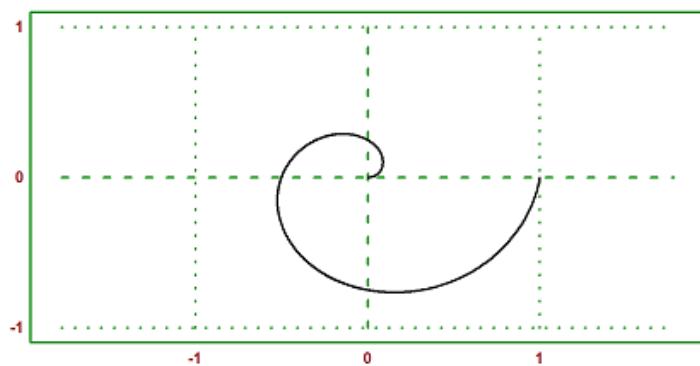
Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.

Dalam contoh berikut, kami memplot spiral

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

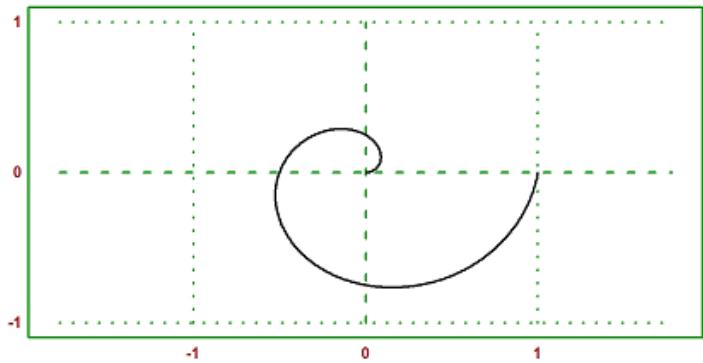
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

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

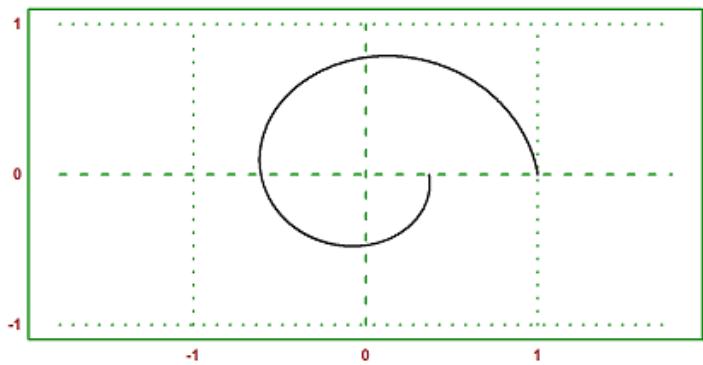


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
>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(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```



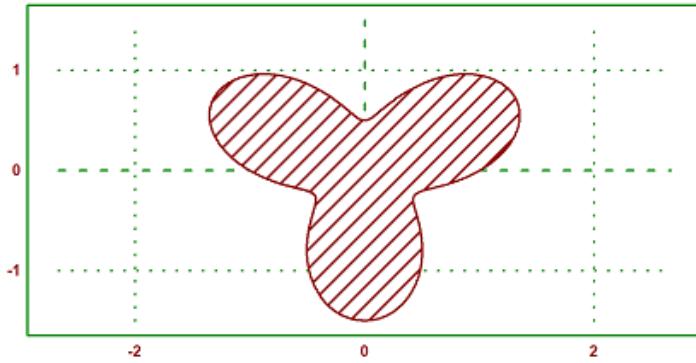
Dalam contoh berikutnya, kami memplot kurva

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

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):
```



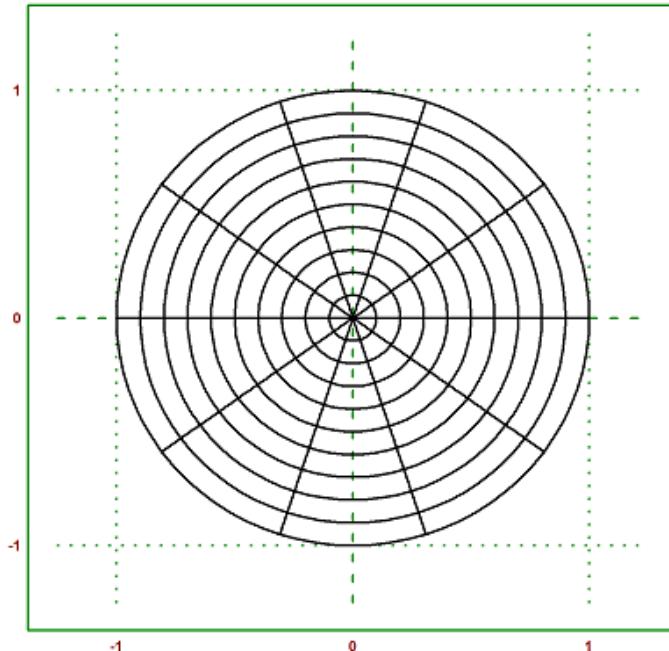
## Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi  $1 \times 2$ ) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

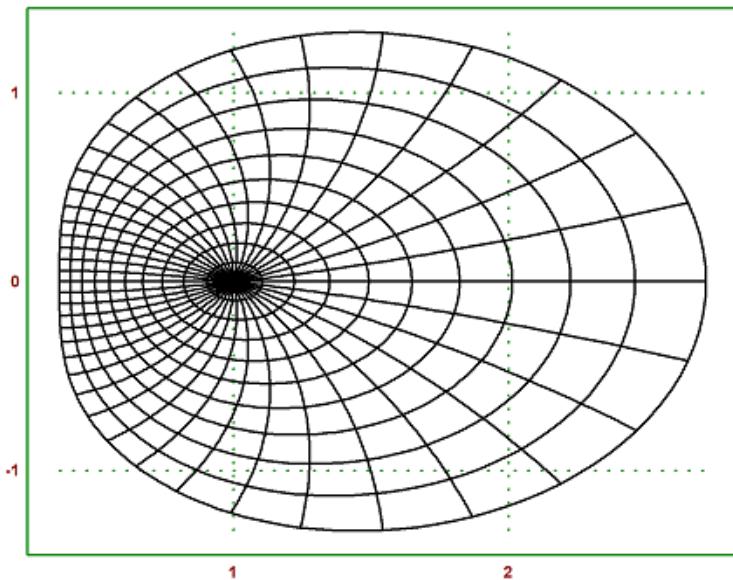
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

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

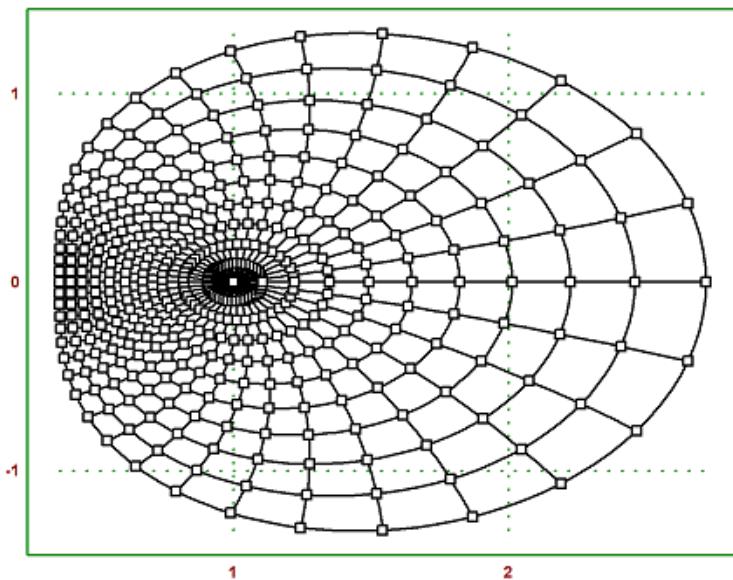
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



```
>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):
```

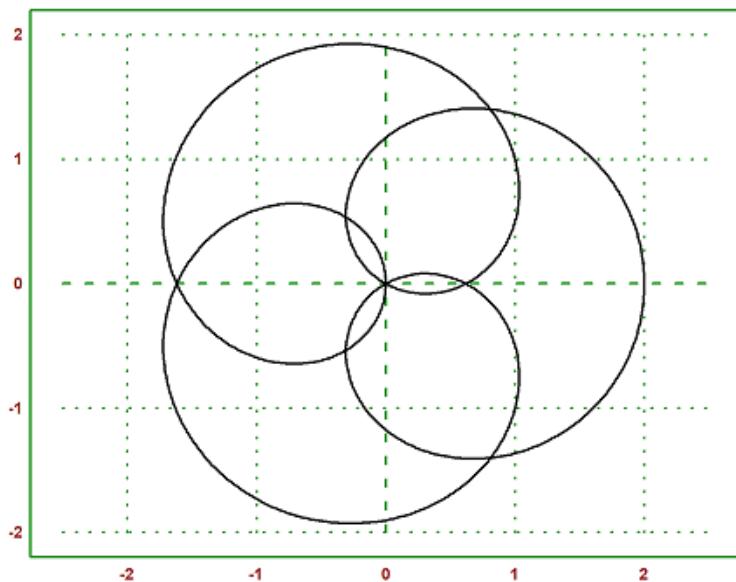


Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

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):
```



## Plot Statistik

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

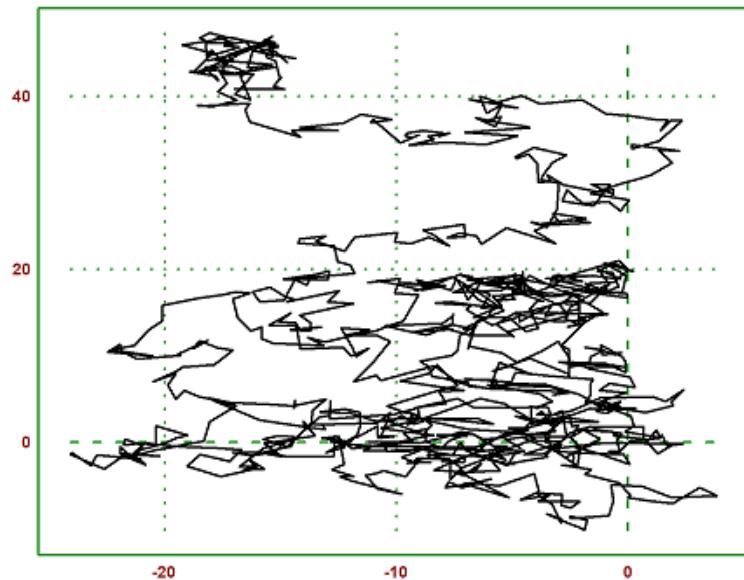
Jumlah kumulatif dari nilai terdistribusi 0-1-normal menghasilkan jalan acak.

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

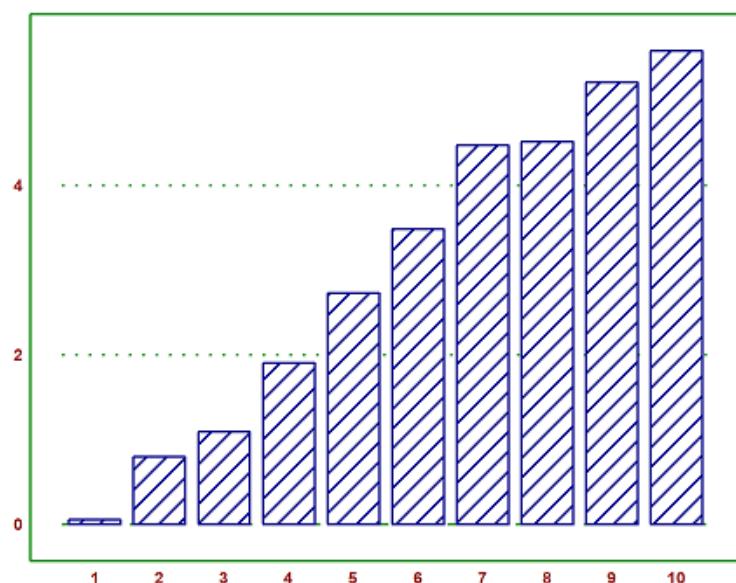


Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

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

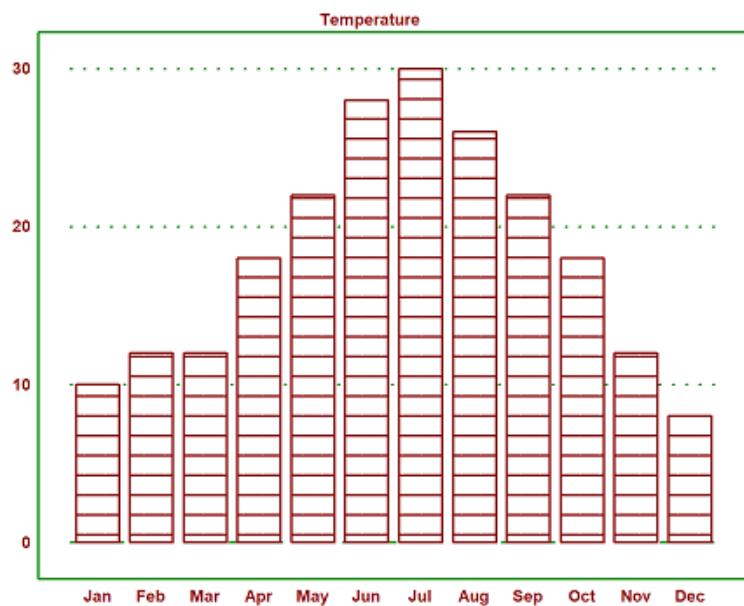


```
>columnsplot(cumsum(random(10)), style="/", color=blue) :
```

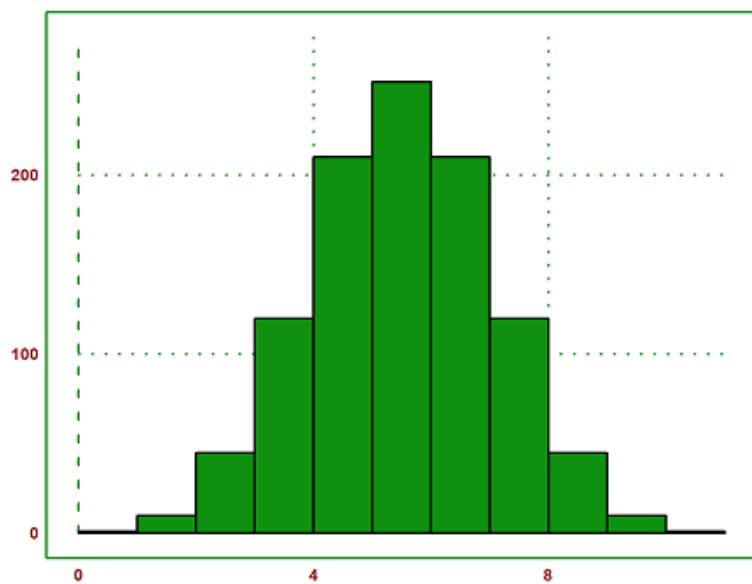


Itu juga dapat menampilkan string sebagai label.

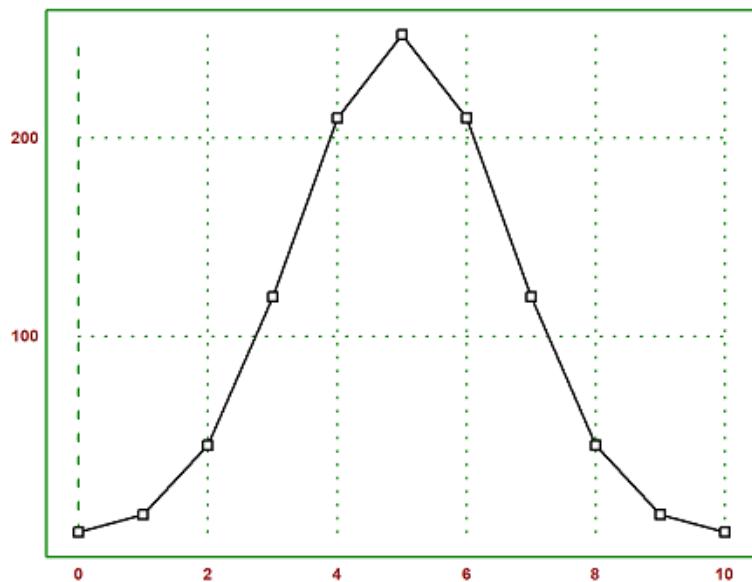
```
>months=["Jan", "Feb", "Mar", "Apr", "May", "Jun", ...
> "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values, lab=months, color=red, style="-");
>title("Temperature"):
```



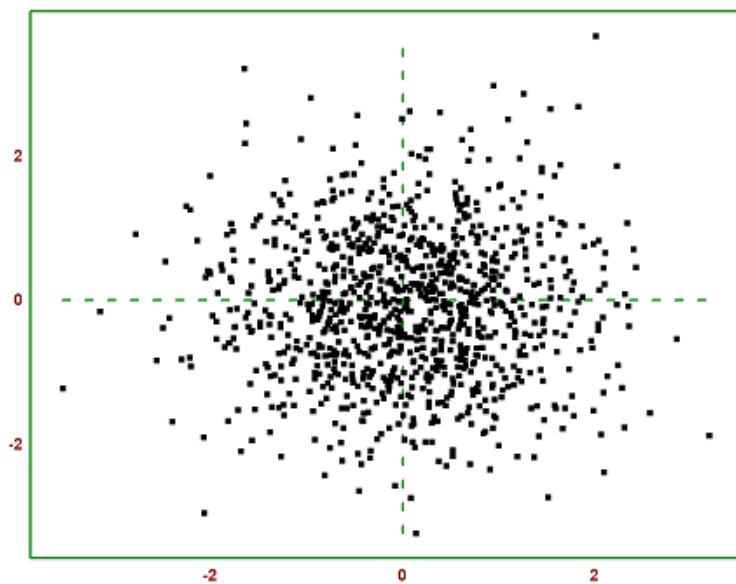
```
>k=0:10;  
>plot2d(k,bin(10,k),>bar):
```



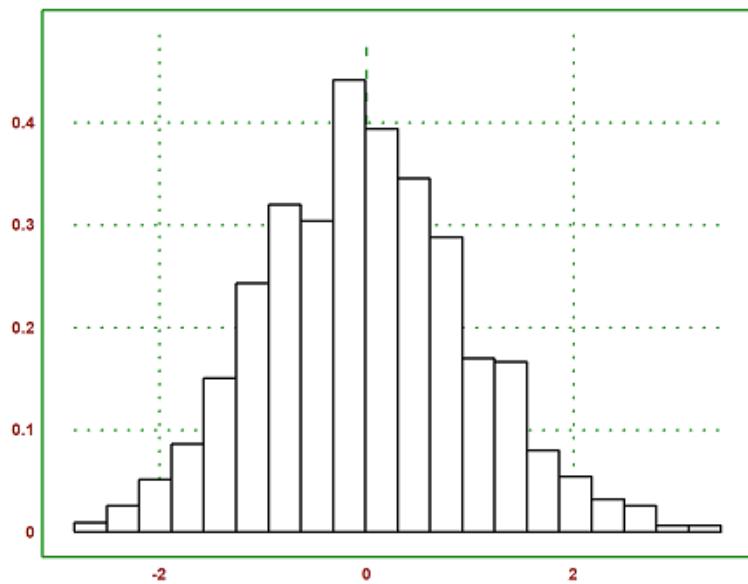
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



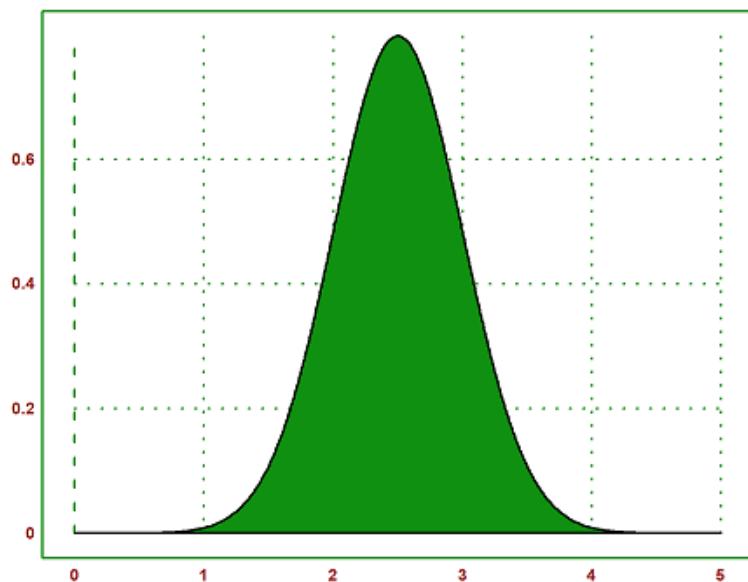
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style="."):
```



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

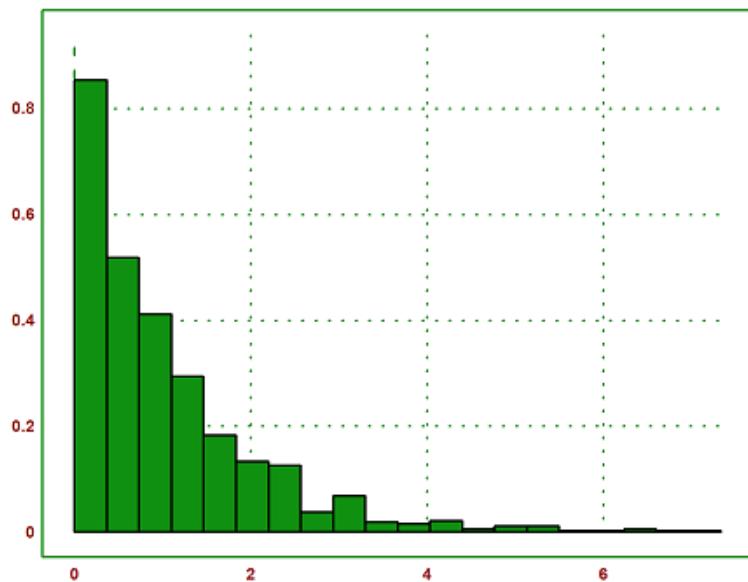


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



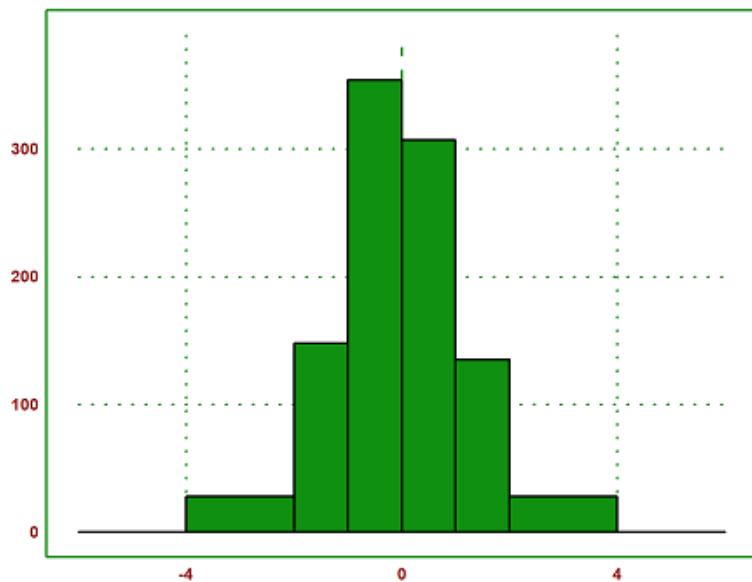
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



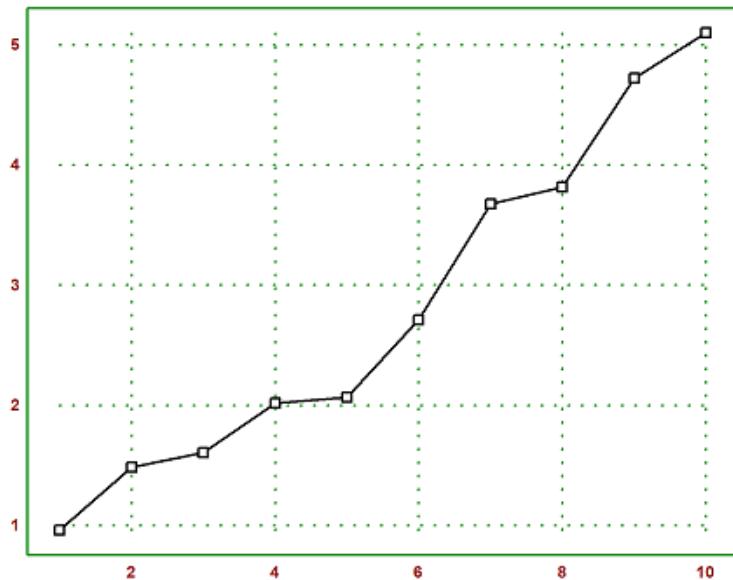
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>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):
```

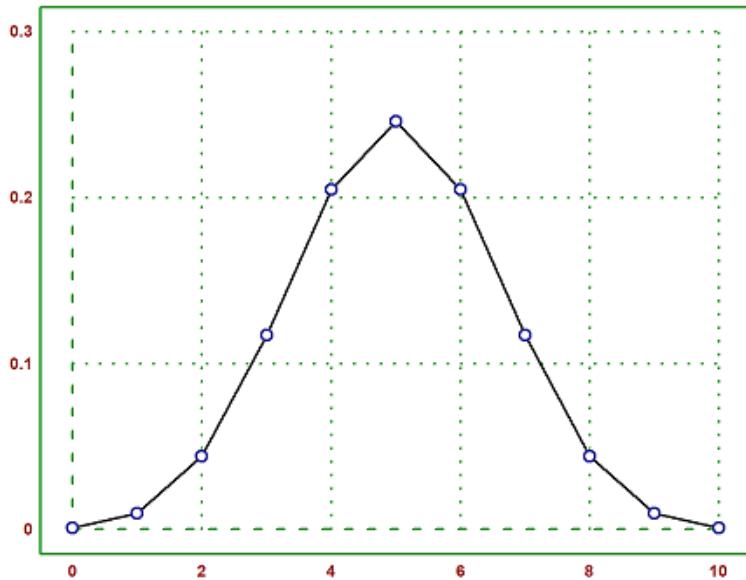


Fungsi statplot() menyetel gaya dengan string sederhana.

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



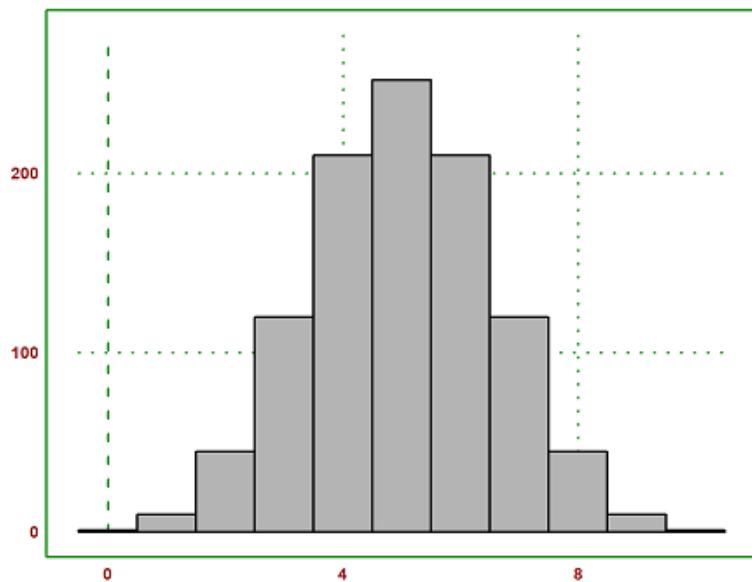
```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



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.

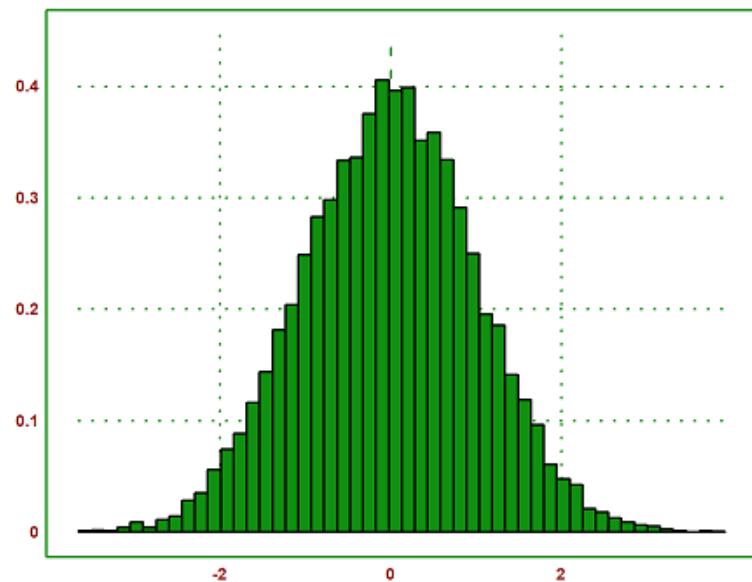
Gaya isian dapat digunakan seperti di atas.

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

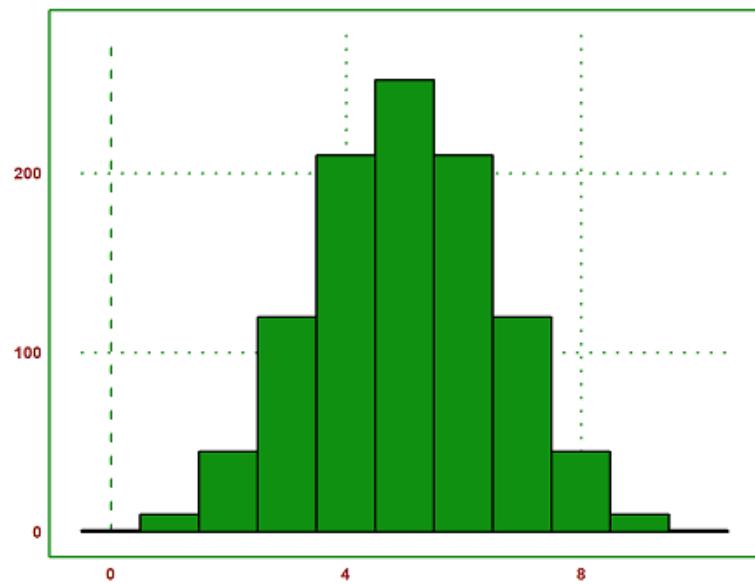


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat dinyatakan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >genap ditentukan, nilai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

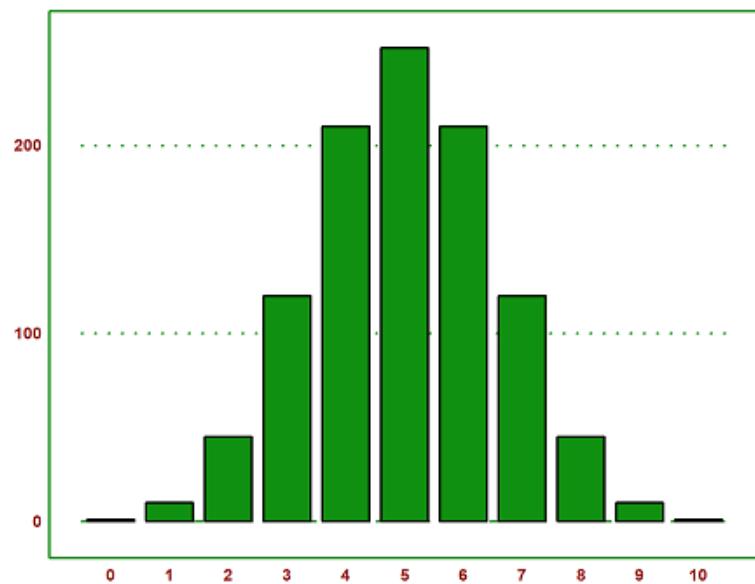
```
>plot2d(normal(10000),distribution=50):
```



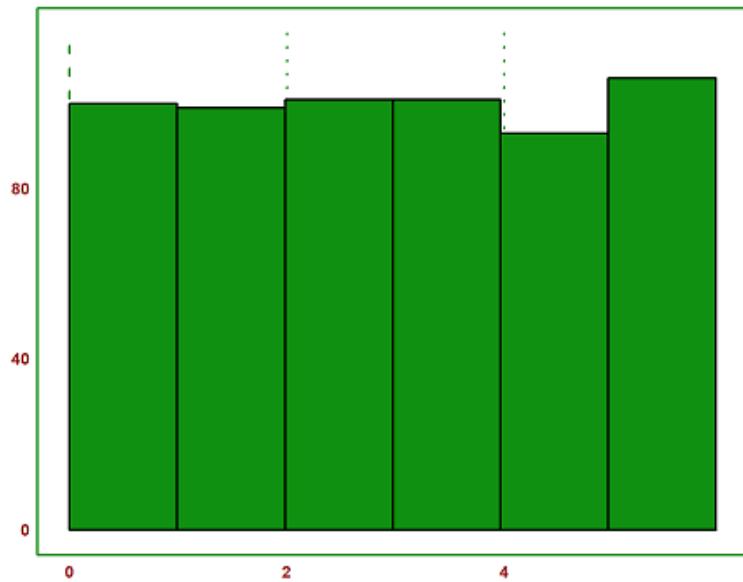
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



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

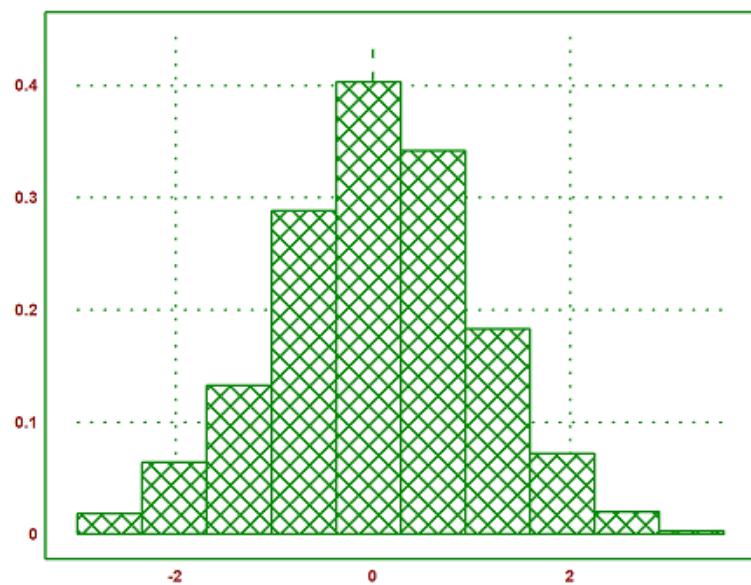


```
>plot2d(random(600)*6, histogram=6) :
```



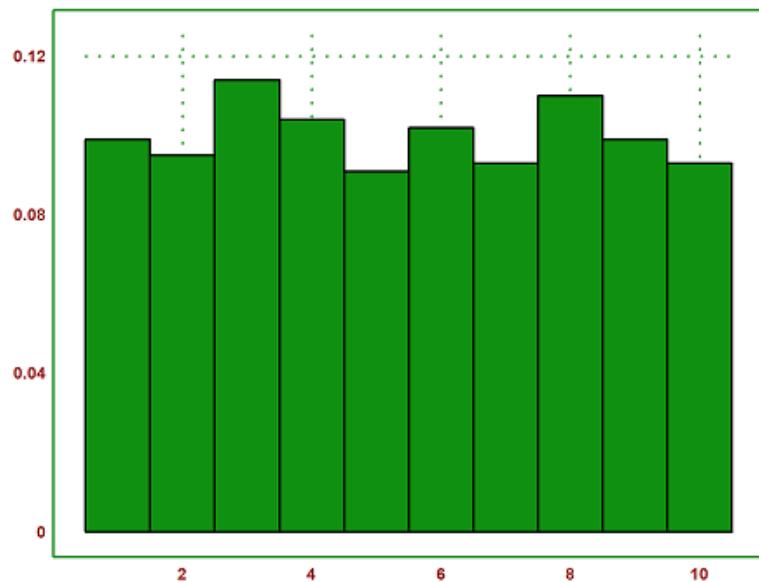
Untuk distribusi, ada parameter distribusi=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



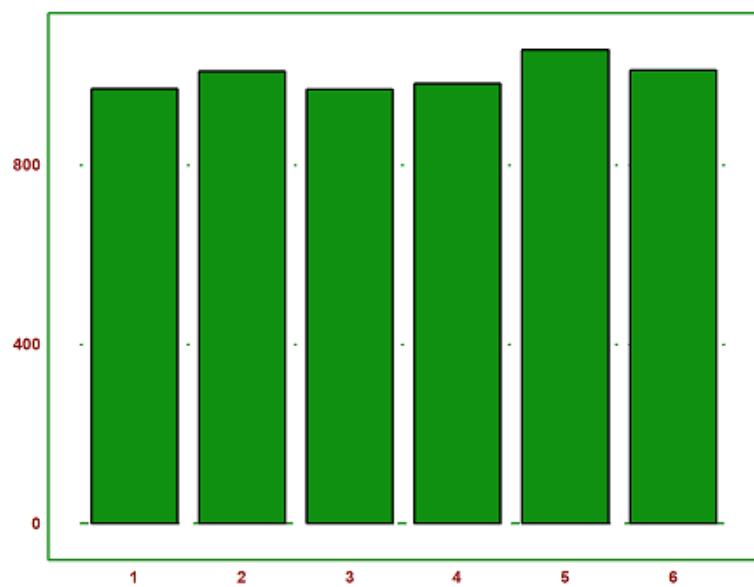
Dengan parameter even=true, ini akan menggunakan interval integer.

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

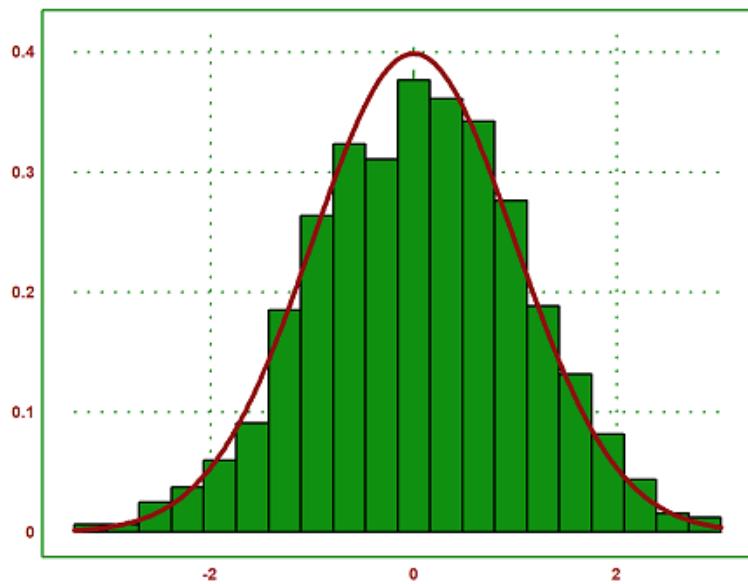


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

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

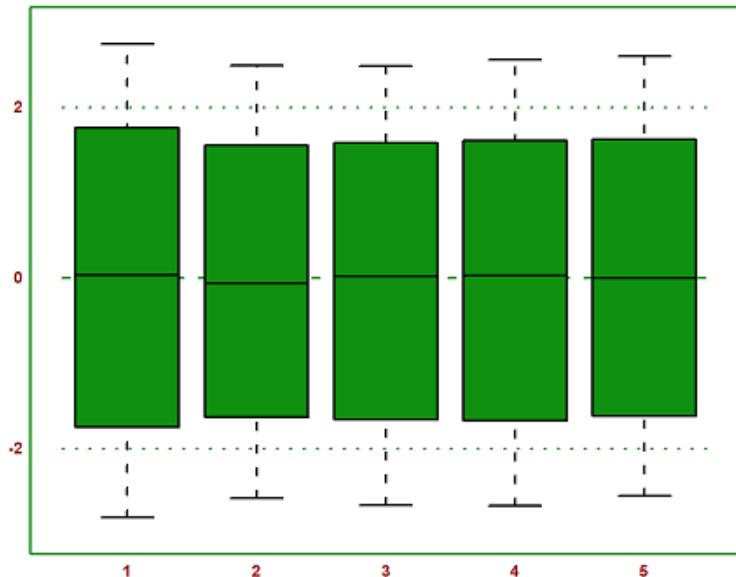


```
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

```
>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 nc, yang akan menyebarkan 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.

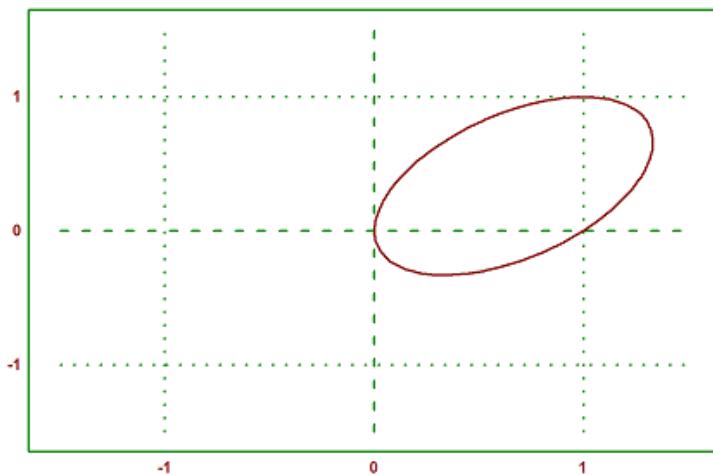
Euler dapat menandai garis level

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

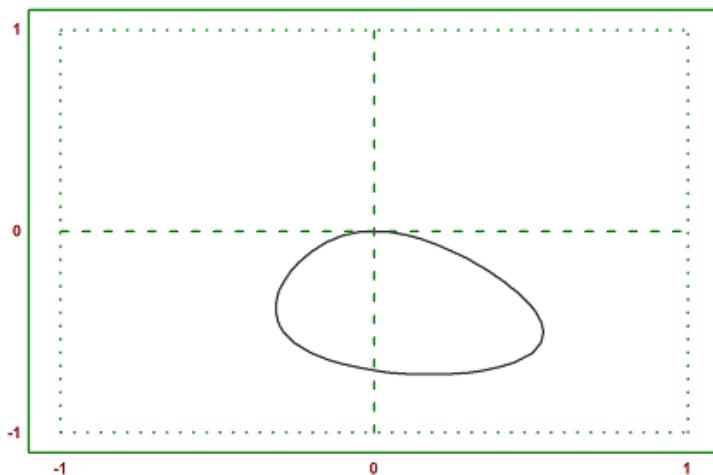
dari fungsi apapun.

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 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.

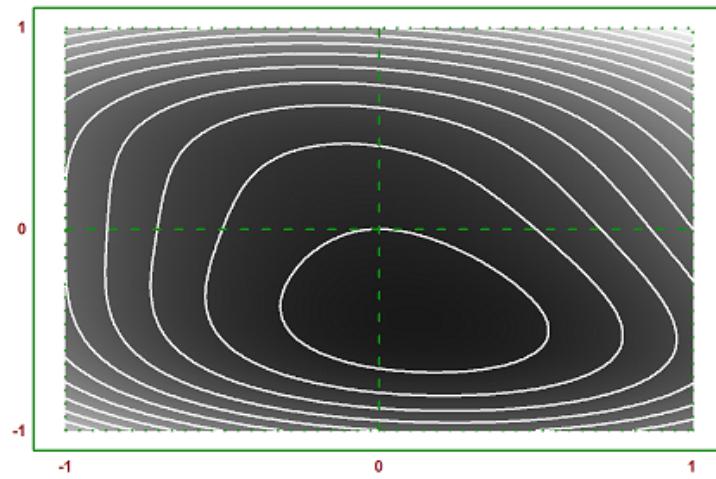
```
>aspect(1.5);  
>plot2d("x^2+y^2-x*y-x", r=1.5, level=0, contourcolor=red):
```



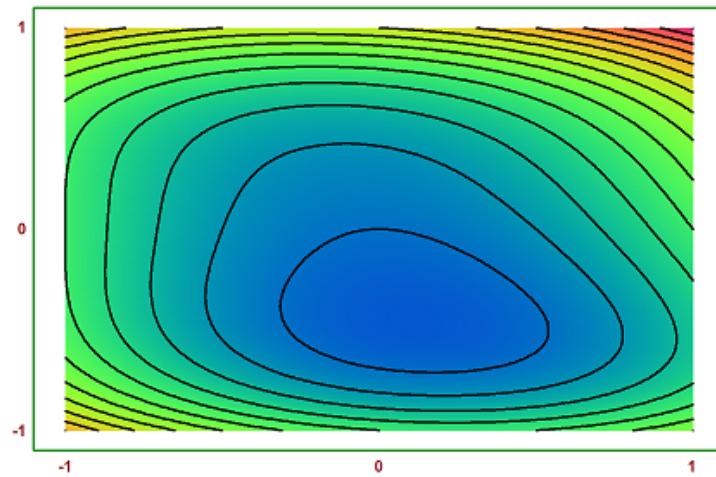
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)  
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```



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

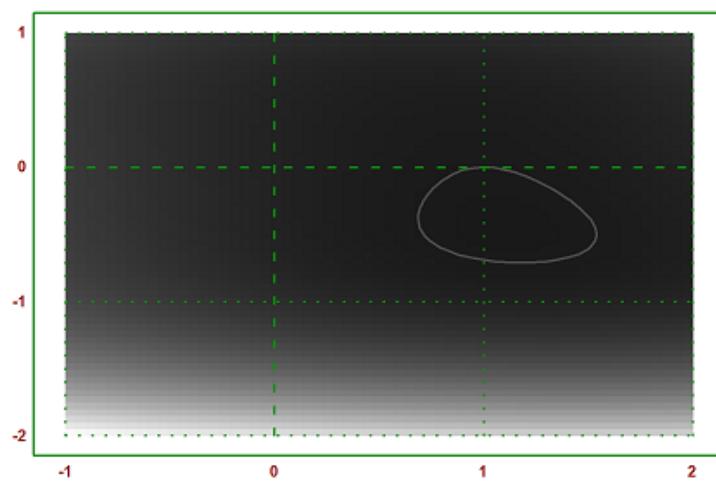


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

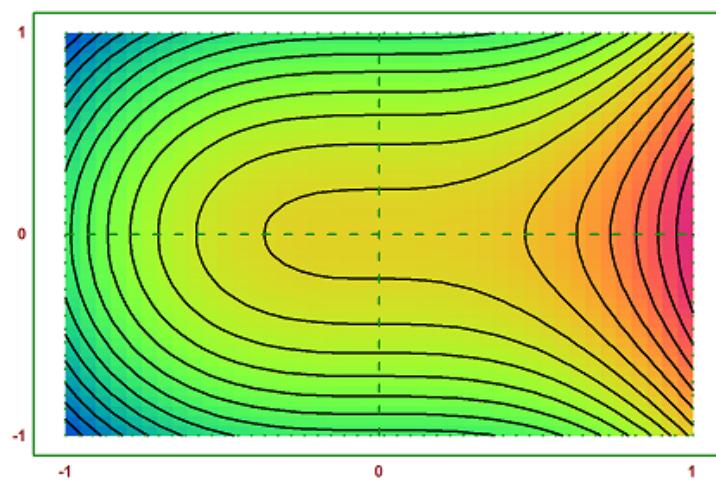


Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

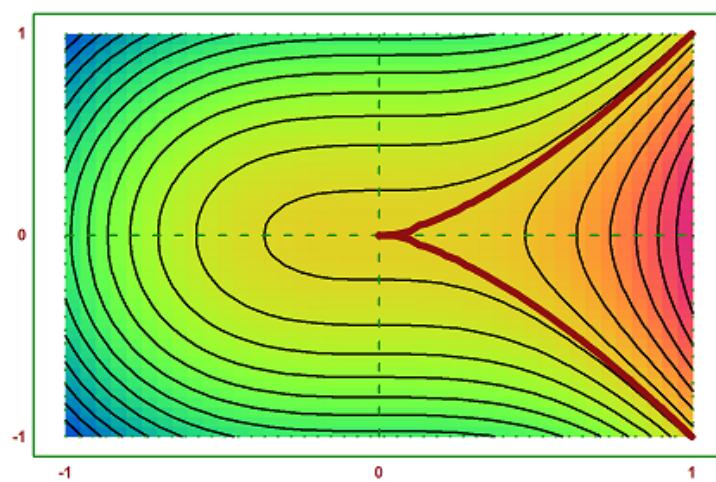
```
>x=-2:0.05:1; y=x'; z=expr(x,y);  
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



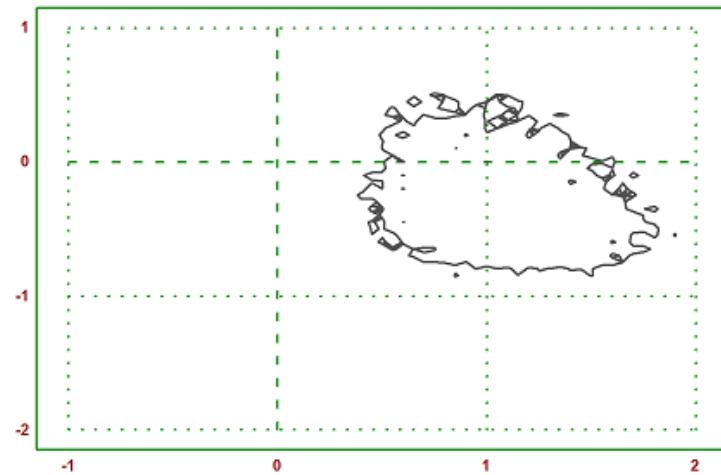
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



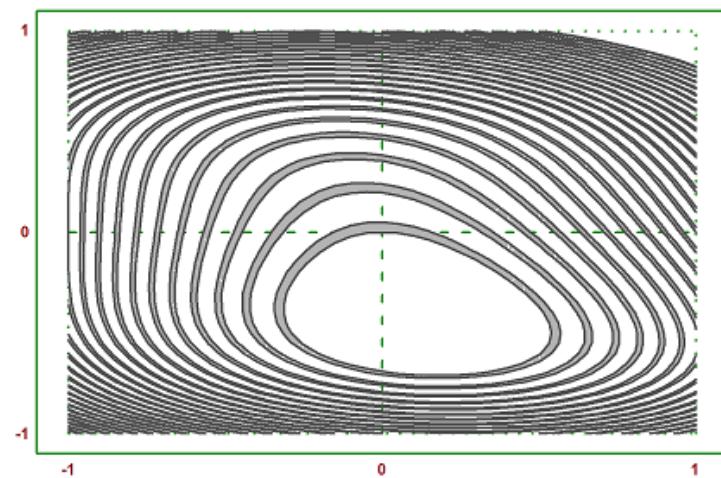
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



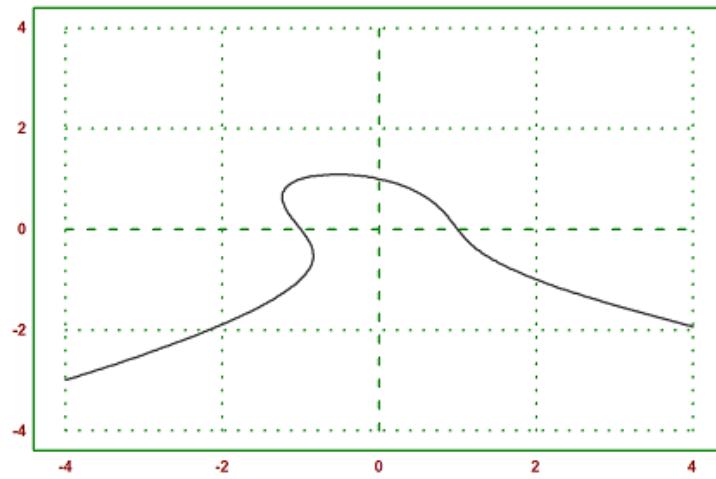
```
>z=z+normal(size(z))*0.2;  
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



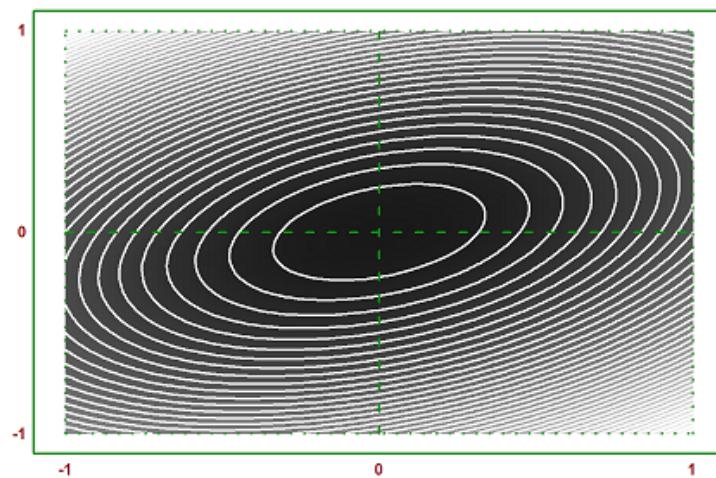
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



```
>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):
```



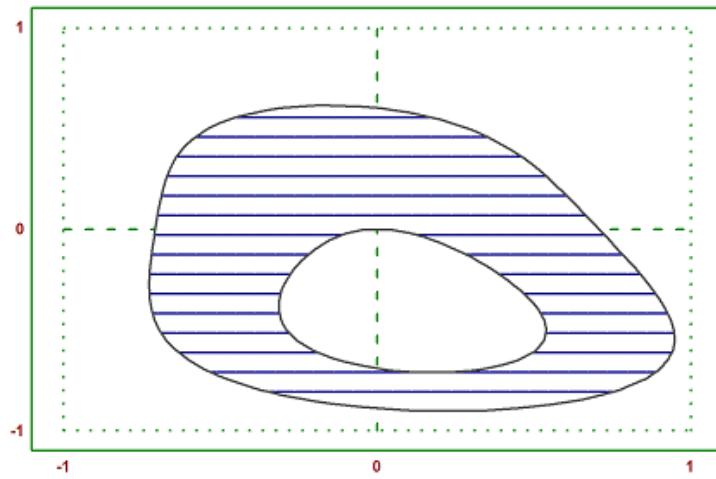
Juga dimungkinkan untuk mengisi set

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

dengan rentang tingkat.

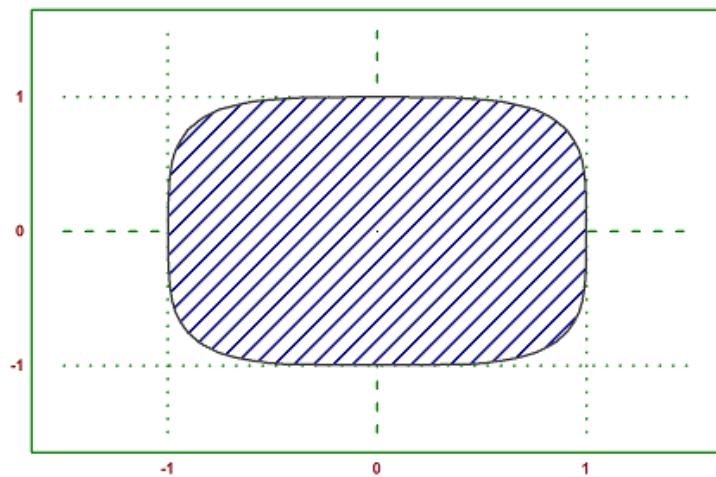
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

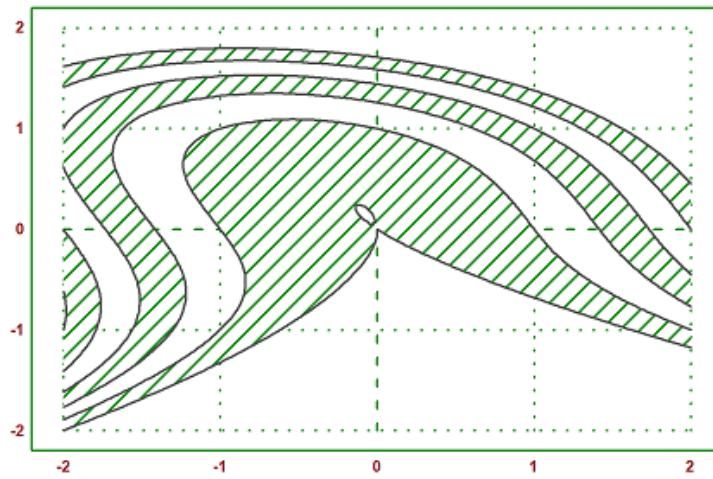


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter `dl` memperluas nilai level ke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

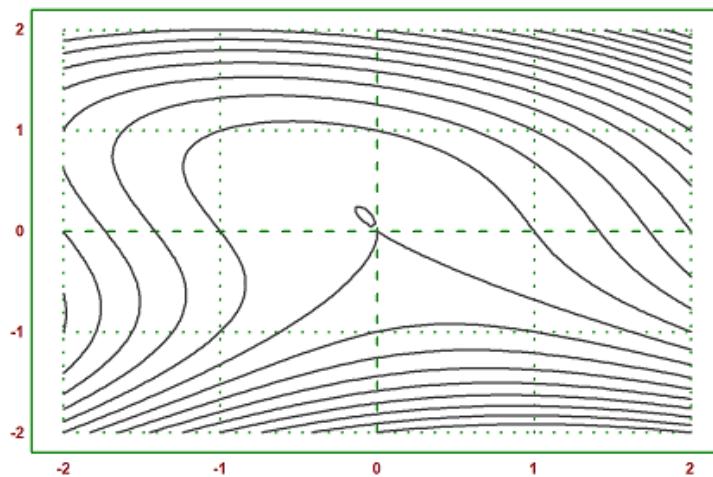
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



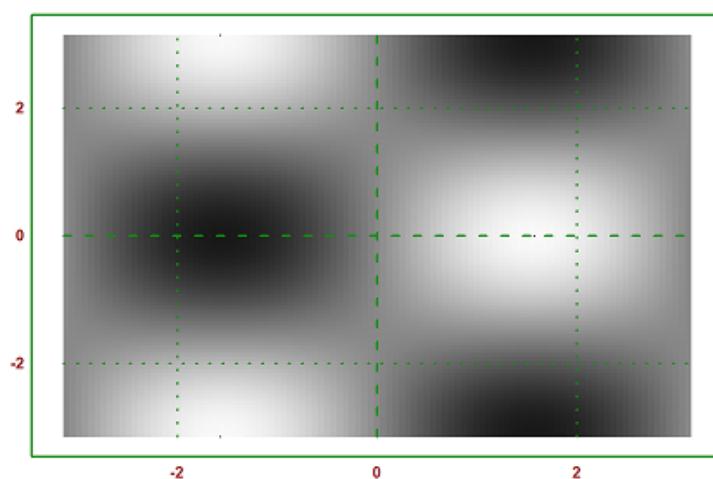
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>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):
```

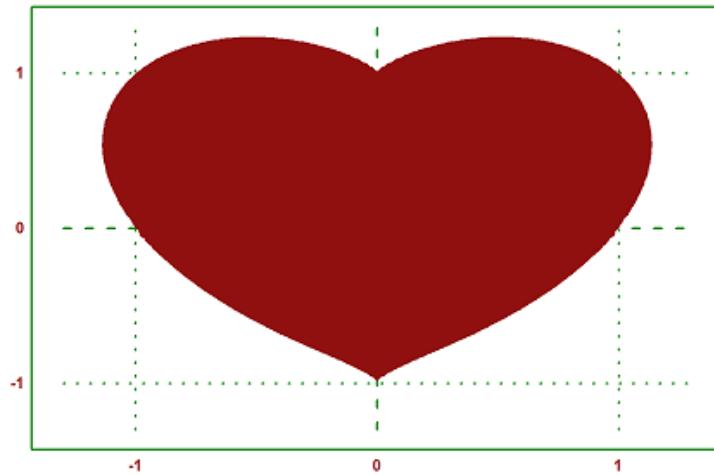


Dimungkinkan juga untuk menandai suatu wilayah

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

Ini dilakukan dengan menambahkan level dengan dua baris.

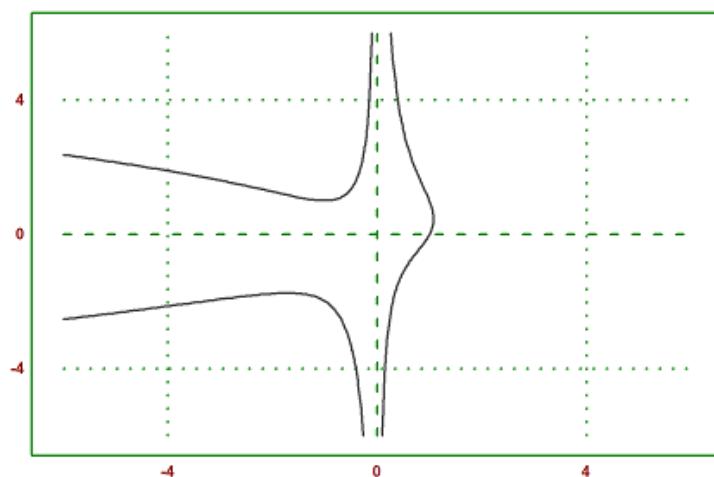
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



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

```

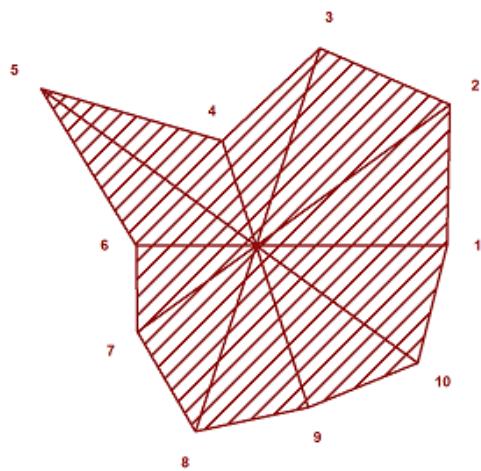
Tidak ada kotak atau sumbu kutu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.

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):

```

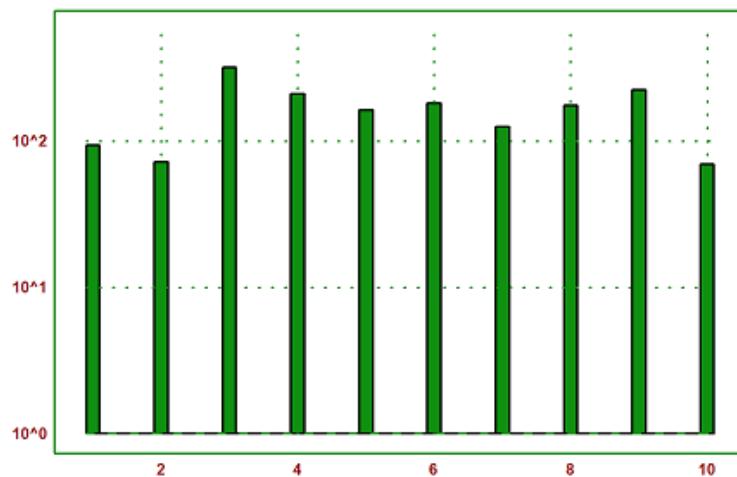


Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```
>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.

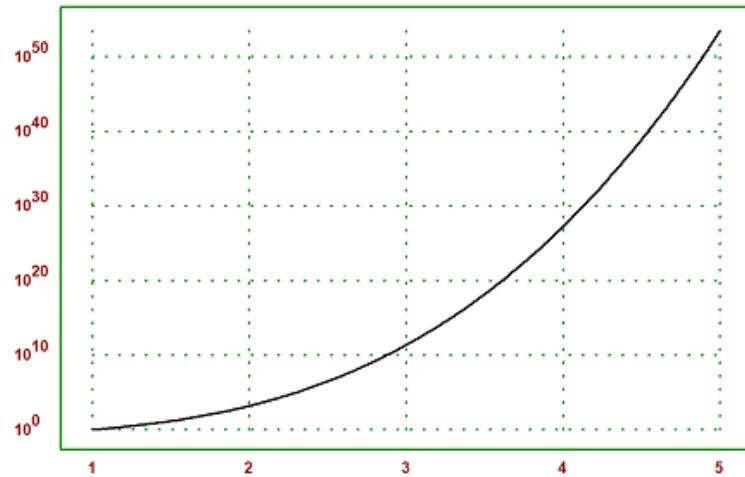
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

## Plot Logaritmik

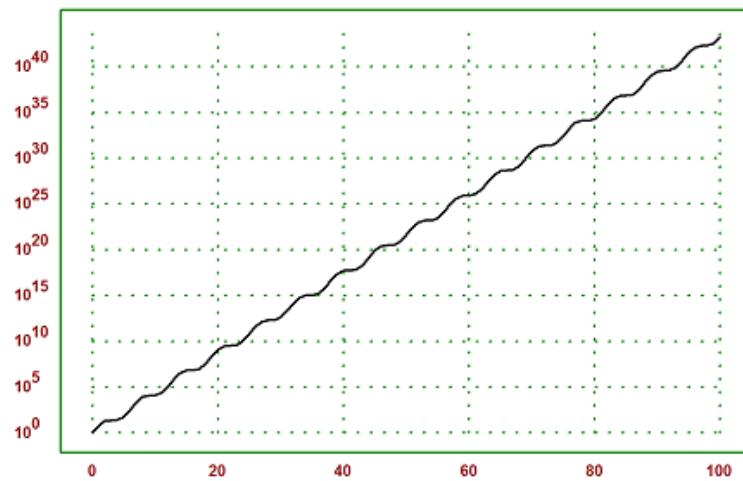
EMT menggunakan parameter "logplot" untuk skala logaritmik. Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

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

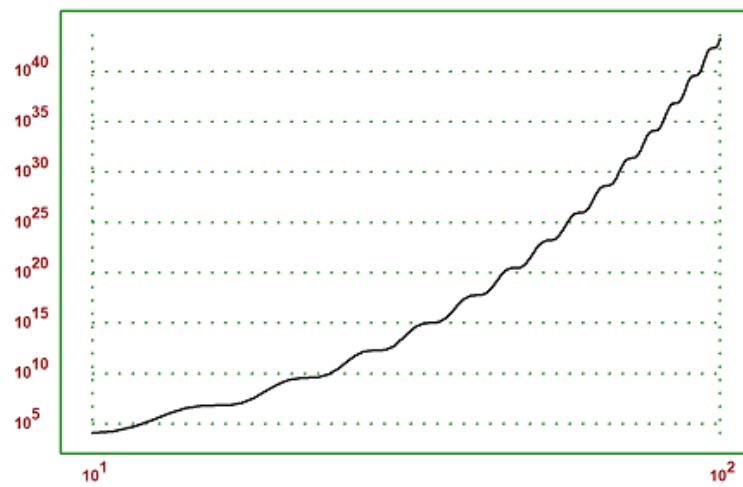
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



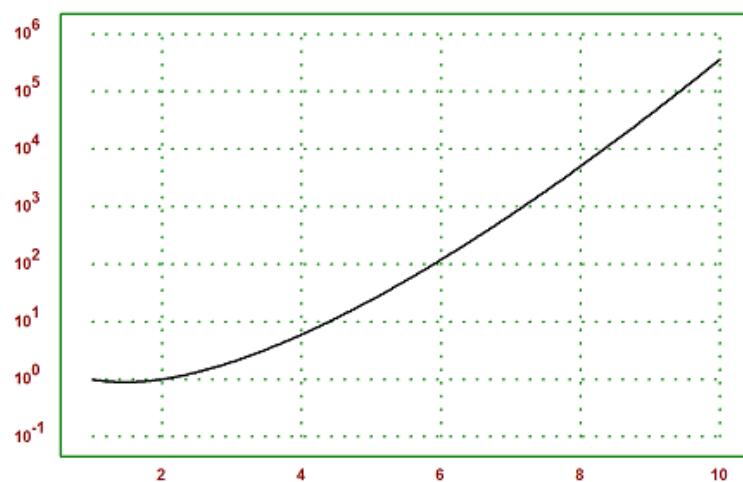
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



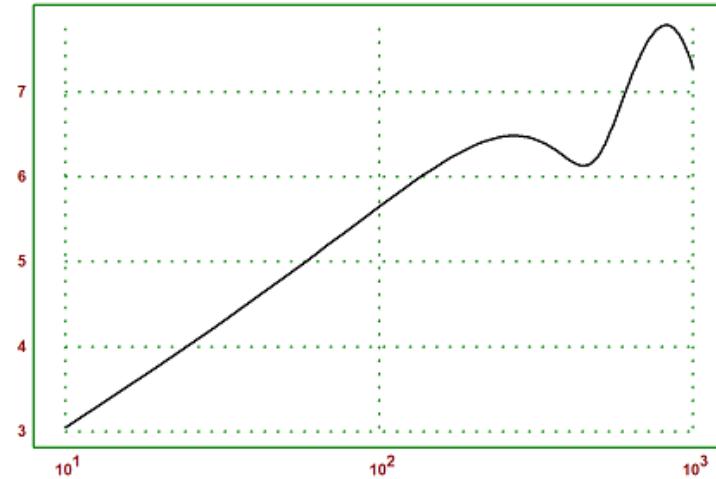
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



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

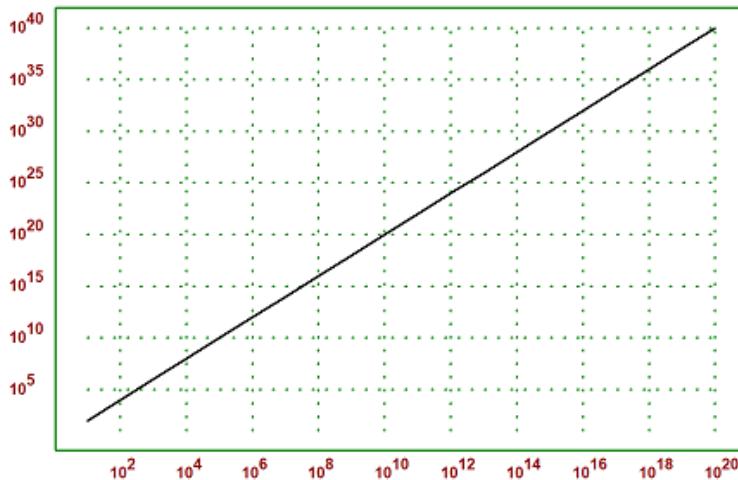


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan plot data.

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



## 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  $2xn$ , 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 [-2,2] 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`.

---

---

## BAB 4

---

# KB PEKAN 7-8: MENGGUNAKAN EMT UNTUK MENGAMBAR GRAFIK 3 DIMENSI (3D)

### Menggambar Plot 3D dengan EMT Ini adalah pengenalan plot 3D

Euler. Kami membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

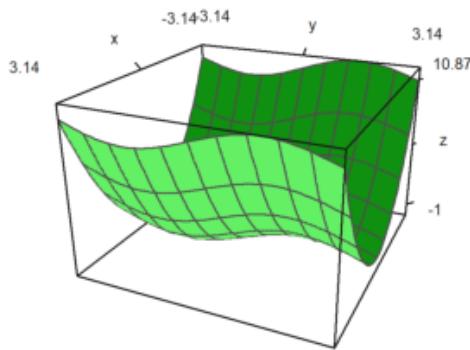
Euler menggambar fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah asal  $x = y = z = 0$ , tetapi sudut = 0 ? melihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler bisa merencanakan

- permukaan dengan bayangan dan garis level atau rentang level,
- awan poin,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari suatu fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur kisaran plot sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)",r=pi):
```



## Fungsi Dua Variabel

Untuk grafik suatu fungsi, gunakan

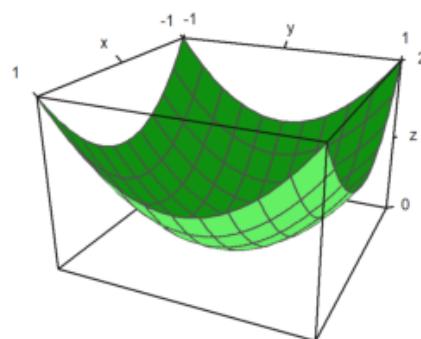
- ekspresi sederhana dalam  $x$  dan  $y$ ,
- nama fungsi dari dua variabel l
- atau matriks data.

Defaultnya adalah kisi kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval kisi adalah 10, tetapi plot menggunakan jumlah default persegi panjang  $40 \times 40$  untuk membuat permukaan. Ini bisa diubah.

- $n = 40, n = [40,40]$ : jumlah garis grid di setiap arah
- $\text{grid} = 10, \text{grid} = [10,10]$ : jumlah garis kisi di setiap arah.

Kami menggunakan default  $n = 40$  dan  $\text{grid} = 10$ .

```
>plot3d("x^2+y^2"):
```



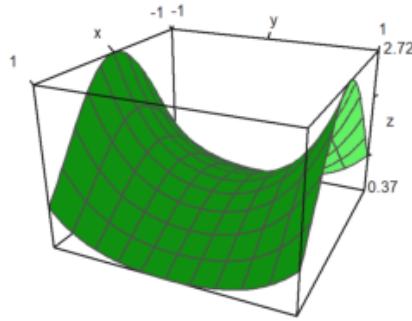
Interaksi pengguna dimungkinkan dengan > user parameter. Pengguna dapat menekan tombol berikut.

- left,right,up,down: putar sudut pandang
- +, -: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih memutar sumber cahaya (lihat di bawah)

- space: reset ke default
- return: interaksi akhir

```
>plot3d("exp(-x^2+y^2)",>user, ...
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



Rentang plot untuk fungsi dapat ditentukan dengan

- a, b: rentang-x
- c, d: rentang y
- r: persegi simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

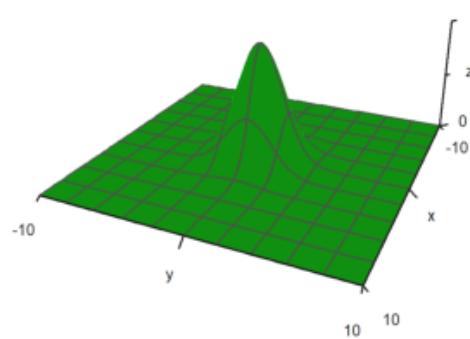
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale>).

scale: angka atau vektor 1x2 untuk skala ke arah x dan y.

frame: jenis bingkai (default 1)

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



Tampilan dapat diubah dengan berbagai cara.

- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut ke sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

Nilai default bisa diperiksa atau diubah dengan fungsi view (). Ini mengembalikan parameter dalam urutan di atas.

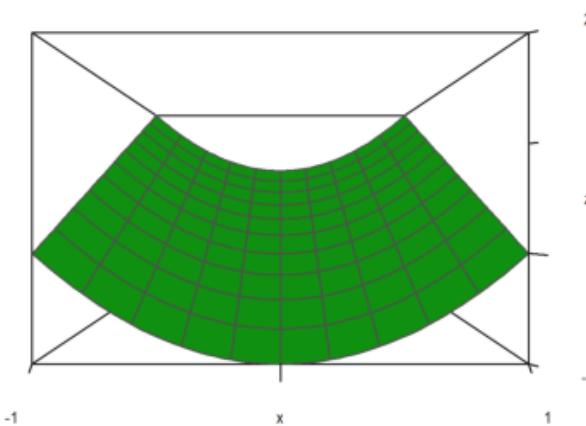
```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar.

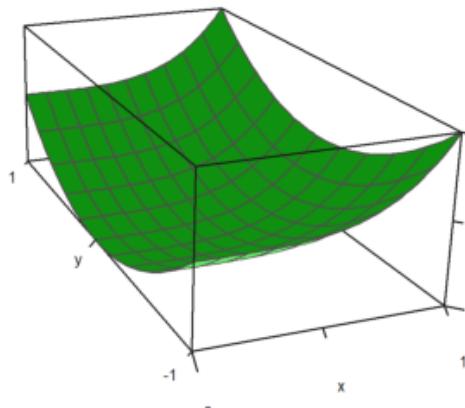
Dalam contoh berikut, sudut = 0 dan tinggi = 0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=2,angle=0,height=0):
```



Plot selalu terlihat ke tengah plot kubus. Anda dapat memindahkan pusat dengan parameter tengah.

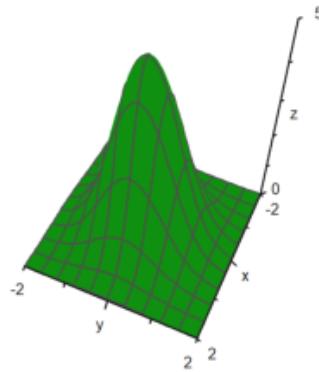
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
> center=[0.4,0,0],zoom=5):
```



Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung ukuran plot. Namun, label mengacu pada ukuran sebenarnya.

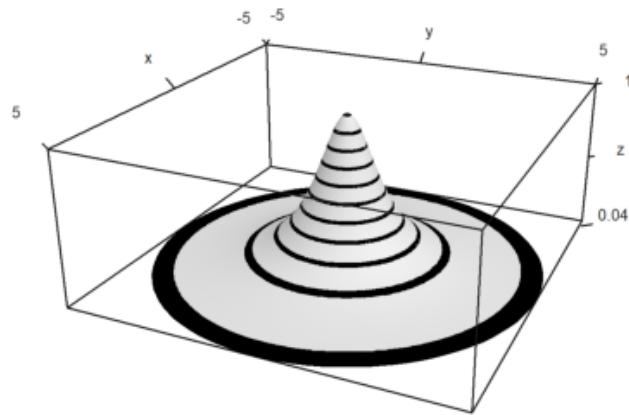
Jika Anda mematikannya dengan `scale = false`, Anda harus berhati-hati, bahwa plot masih pas dengan jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan pusat.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
> center=[0,0,-2],frame=3):
```

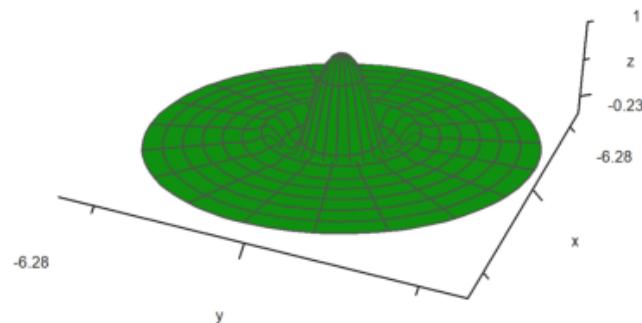


Plot kutub juga tersedia. The parameter `polar= true` menggambarkan plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

```
>plot3d("1/ (x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```



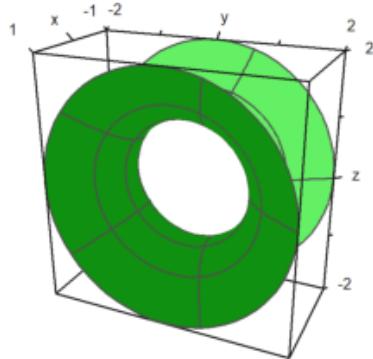
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```



Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

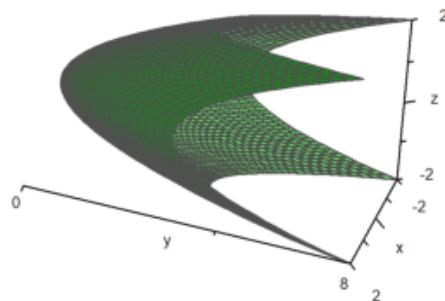
- rotate = 1: Menggunakan sumbu x
- rotate = 2: Menggunakan sumbu z

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



Berikut adalah plot dengan tiga fungsi.

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3) :
```



## Plot Kontur

Untuk plot, Euler menambahkan garis kisi. Alih-alih, dimungkinkan untuk menggunakan garis level dan corak satu warna atau corak warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan shading. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.

-> hue: Mengaktifkan bayangan terang, bukan kabel.

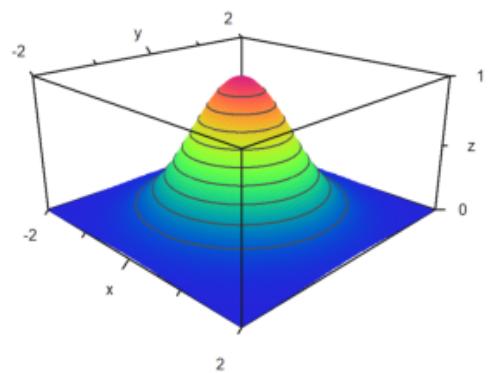
-> contour: Membuat plot garis kontur otomatis pada plot.

- level = ... (atau level): Vektor nilai untuk garis kontur.

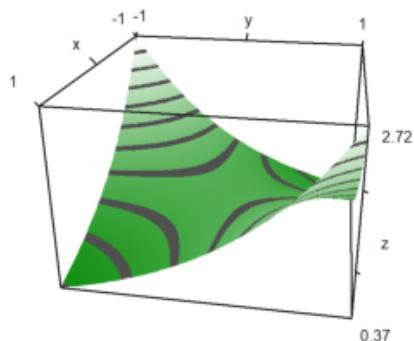
Standarnya adalah level = "auto", yang menghitung beberapa baris level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan kisi yang lebih halus untuk 100x100 titik, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)", r=2, n=100, level="thin", ...
>>contour, >spectral, fscale=1, scale=1.1, angle=45°, height=20°) :
```



```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```



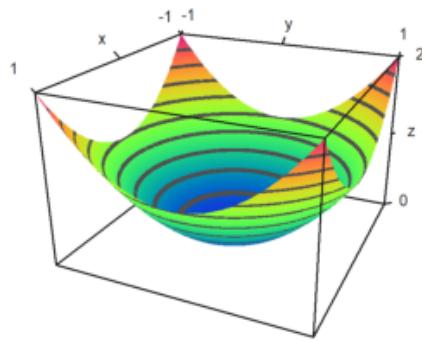
Bayangan default menggunakan warna abu-abu. Tetapi berbagai spektrum warna juga tersedia.

-> spectral: Menggunakan skema spektral default

- color = ...: Menggunakan warna khusus atau skema spektral

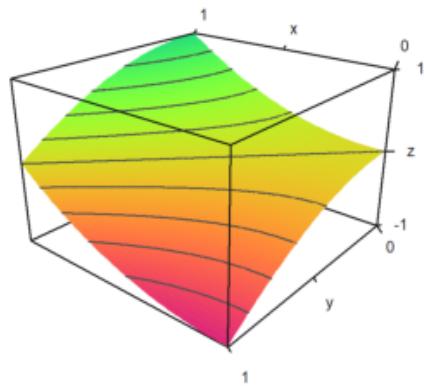
Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah poin untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



Alih-alih garis level otomatis, kami juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis, bukan rentang level.

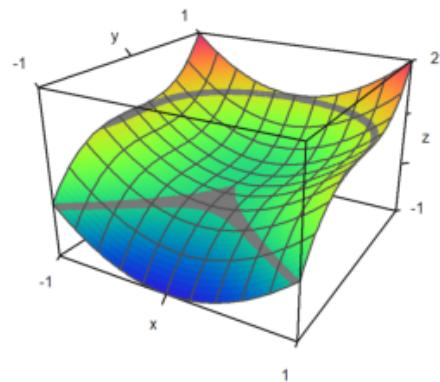
```
>plot3d("x^2-y^2", 0, 1, 0, 1, angle=220°, level=-1:0.2:1, color=redgreen) :
```



Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.

Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3", level=[-0.1,0.9;0,1], ...
> >spectral, angle=30°, grid=10, contourcolor=gray) :
```

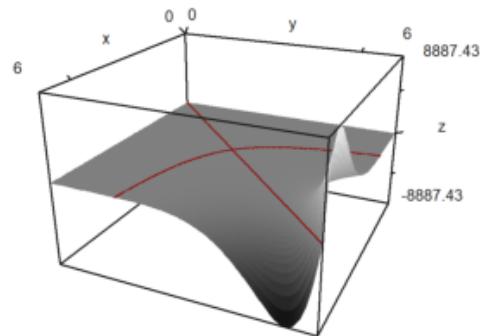


Dalam contoh berikut, kami memplot set, di mana

$$f(x, y) = x^y - y^x = 0$$

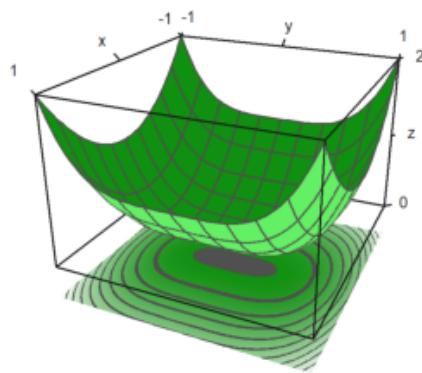
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x", level=0, a=0, b=6, c=0, d=6, contourcolor=red, n=100):
```



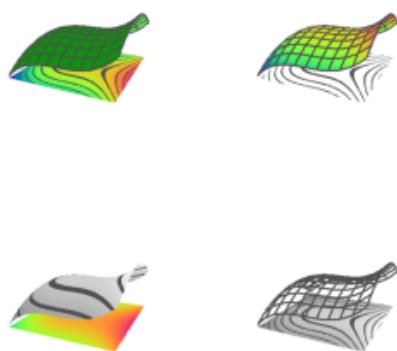
Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4", >cp, cpcolor=green, cpdelta=0.2):
```



Berikut beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1); ...
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2); ...
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3); ...
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4); ...
> plot3d(expr,<frame,>hue,grid=10,>cp,cpcolor=gray); ...
>figure(0):
```



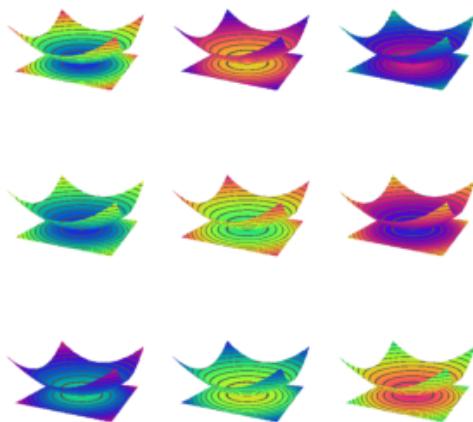
Ada beberapa skema spektral lain, diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana nilai

- spectral: untuk rentang dari biru hingga merah
- white: untuk rentang yang lebih redup
- yellowblue,purplegreen,blueyellow,greenred
- blueyellow, greenpurple,yellowblue,redgreen

```

>figure(3,3); ...
>for i=1:9; ...
> figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4); ...
>end; ...
>figure(0);

```



Sumber cahaya dapat diubah dengan l dan tombol cursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

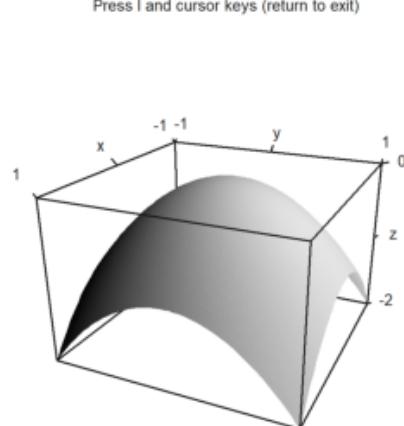
-light : arah cahaya

- amb: cahaya ambient antara 0 dan 1

Perhatikan bahwa program tidak membuat perbedaan antara sisi-sisi plot. Tidak ada bayangan. Untuk ini, Anda membutuhkan Povray.

```

>plot3d("-x^2-y^2", ...
> hue=true,light=[0,1,1],amb=0,user=true, ...
> title="Press l and cursor keys (return to exit)":
```



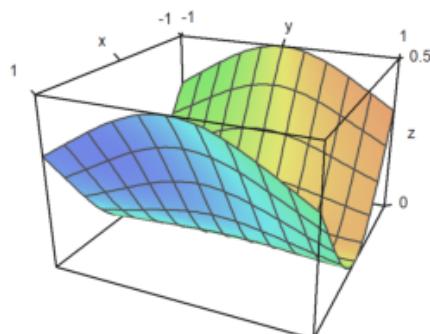
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
> zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```



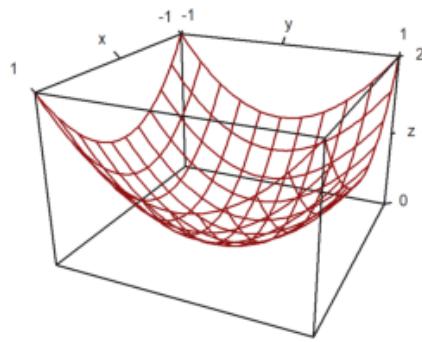
Warna 0 memberikan efek pelangi khusus.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```



## Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur plot3d termasuk plot implisit. Plot ini menunjukkan himpunan nol fungsi dalam tiga variabel.

Solusi dari

$$f(x, y, z) = 0$$

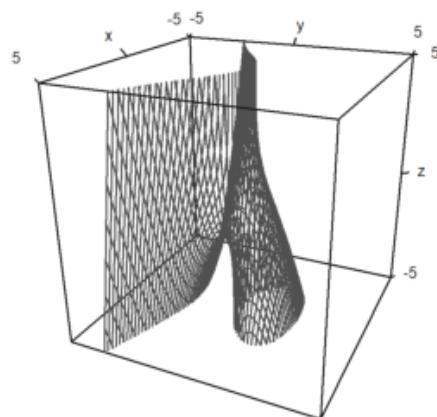
dapat divisualisasikan dalam potongan sejajar dengan bidang x-y-, x-z-, dan y-z.

- implisit = 1: potong sejajar bidang y-z
- implisit = 2: potong sejajar dengan bidang x-z
- implisit = 4: potong sejajar bidang x-y

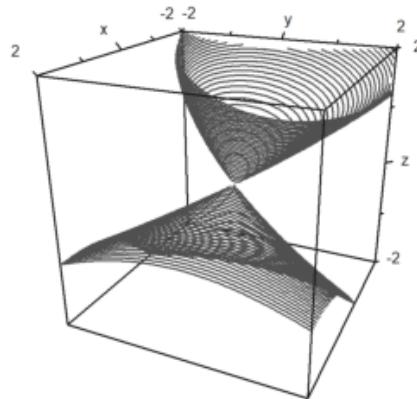
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh yang kami plot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

```
>plot3d("x^2+y^3+z*y-1", r=5, implicit=3) :
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



## Plotting 3D Data

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu memberikan matriks nilai x, y dan z, atau tiga fungsi atau ekspresi  $f_x(x, y)$ ,  $f_y(x, y)$ ,  $f_z(x, y)$ .

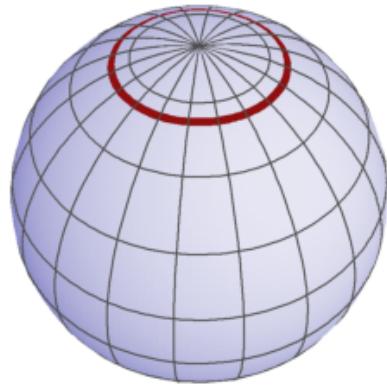
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x, y, z adalah matriks, kita asumsikan bahwa (t, s) berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di luar angkasa.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

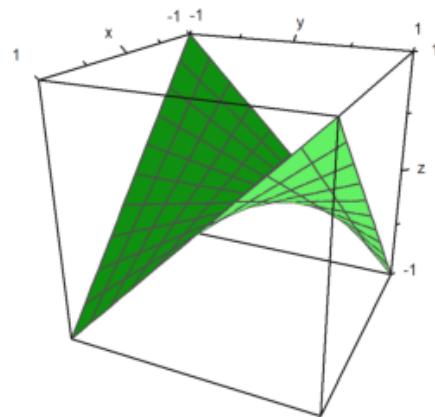
Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita bisa menandai daerah, dalam kasus kita daerah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```



Berikut adalah contoh grafik dari suatu fungsi.

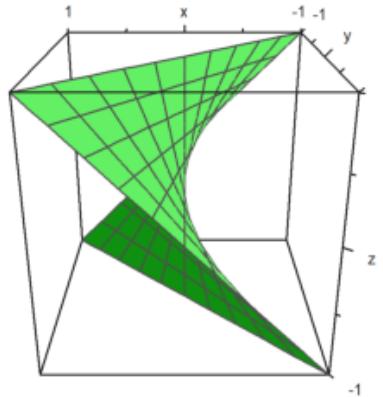
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Namun, kita bisa membuat semua jenis permukaan. Ini adalah permukaan yang sama sebagai suatu fungsi

$$x = y z$$

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```



Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

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

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami menyimpangkan ini dengan sebuah faktor

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

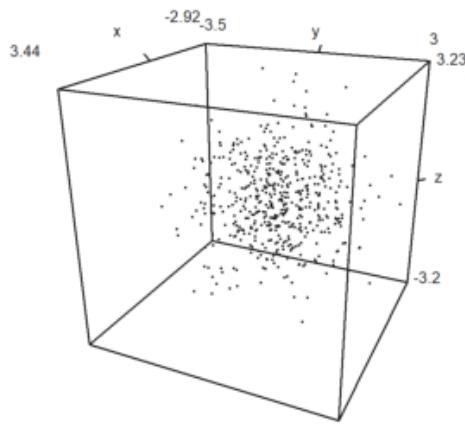
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):
```



Tentu saja, point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor sebagai koordinat titik.

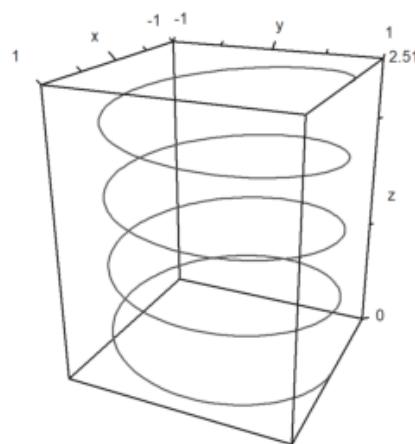
Gayanya sama seperti di plot2d dengan points = true;

```
>n=500; ...
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

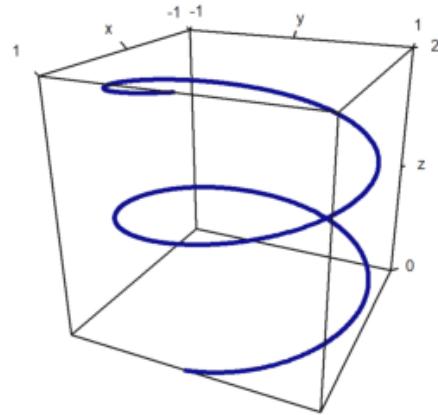


Juga dimungkinkan untuk memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung sebelumnya titik-titik kurva. Untuk kurva di bidang kita menggunakan urutan koordinat dan parameter wire = true.

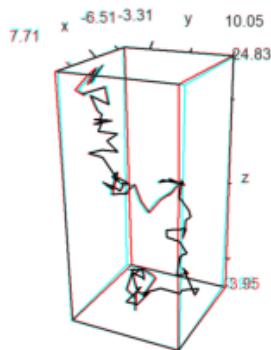
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3,wirecolor=blue):
```

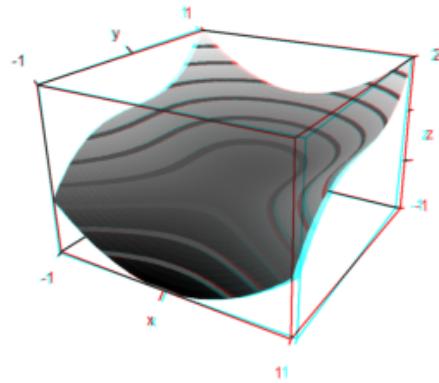


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire) :
```



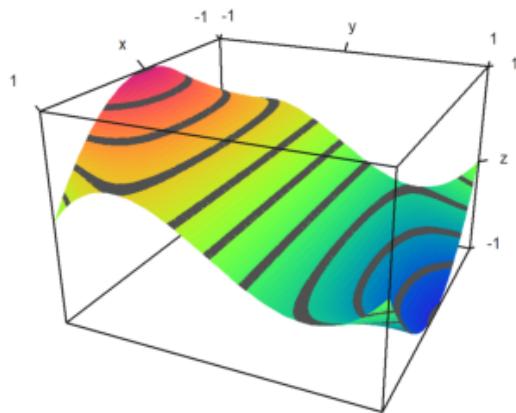
EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda membutuhkan red/cyan glasses.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°) :
```



Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian fungsinya.

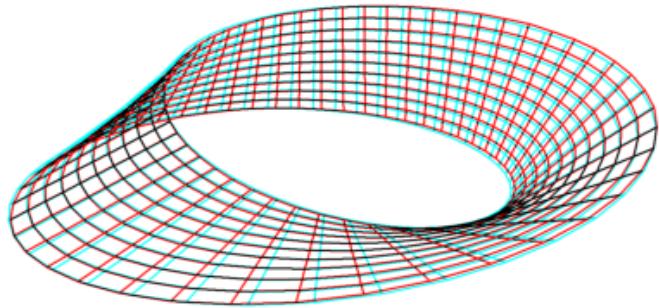
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2) :
```



Euler juga dapat memplot permukaan berparameter, jika parameternya adalah nilai x, y, dan z dari gambar kisi persegi panjang di dalam ruang.

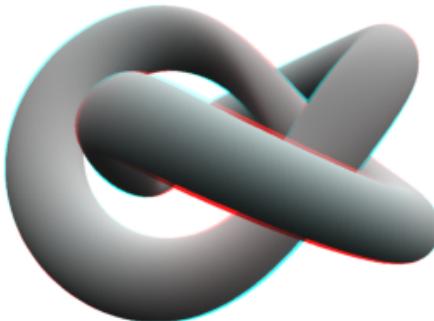
Untuk demo berikut, kami menyiapkan parameter u- dan v-, dan menghasilkan koordinat ruang dari ini.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3) :
```



Berikut adalah contoh yang lebih rumit, yang megah dengan red/cyan glasses.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
>z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



#### \*Plot Statistik

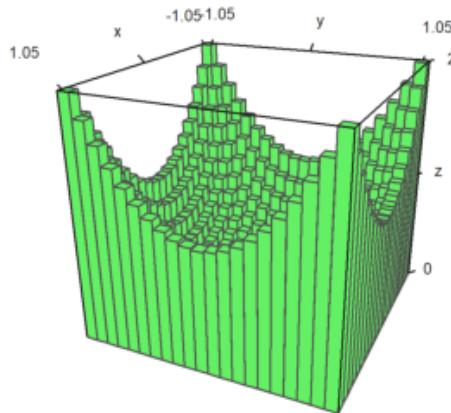
Plot batang juga dimungkinkan. Untuk ini, kami harus menyediakan

- x: vektor baris dengan  $n + 1$  elemen
- y: vektor kolom dengan  $n + 1$  elemen
- z: nxn matriks nilai.

z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.

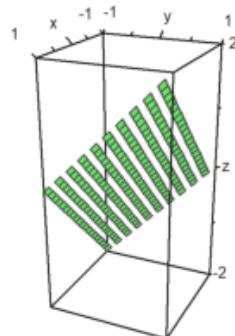
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x+1.1)-0.05; ya=(y-1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```



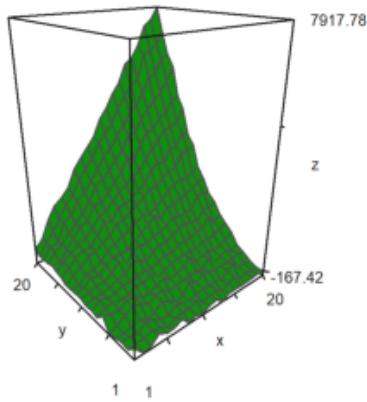
Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```

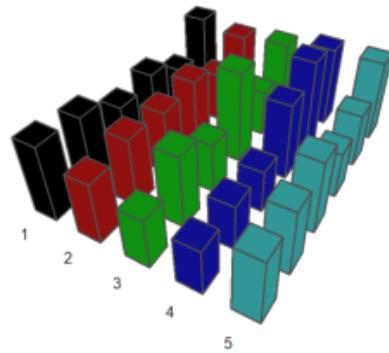


Jika memuat atau membuat matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan skala (M), atau menskalakan matriks dengan >zscale. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8);
```



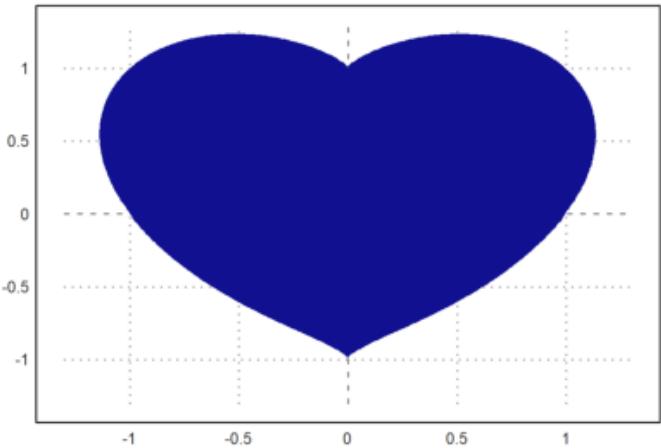
```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



## Permukaan Benda Putar

---

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



```
>ekspressi &= (x^2+y^2-1)^3-x^2*y^3; $ekspressi
```

$$(y^2 + x^2 - 1)^3 - x^2 y^3$$

Kami ingin memutar kurva jantung di sekitar sumbu y. Inilah ungkapan yang mendefinisikan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kita atur

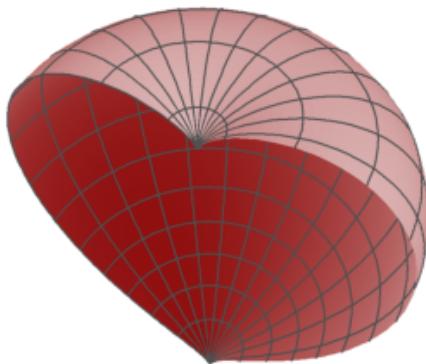
$$x = r \cos(a), \quad y = r \sin(a).$$

```
>function fr(r,a) &= ekspressi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Hal ini memungkinkan untuk menentukan fungsi numerik, yang menyelesaikan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang berubah sebagai permukaan parametrik.

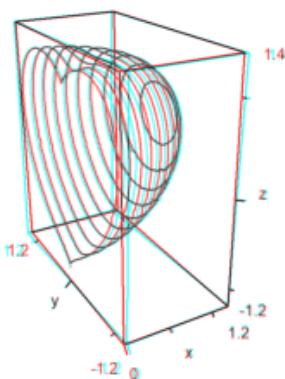
```
>function map f(a) := bisect("fr", 0, 2; a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t); ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang mendeskripsikan objek.

```
>function f(x,y,z) ...  
  
r=x^2+y^2;  
return (r+z^2-1)^3-r*z^3;  
endfunction
```

```
>plot3d("f(x,y,z)", ...  
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...  
>implicit=1,angle=-30°,zoom=2.5,n=[10,60,60],>anaglyph):
```



## Special 3D Plots

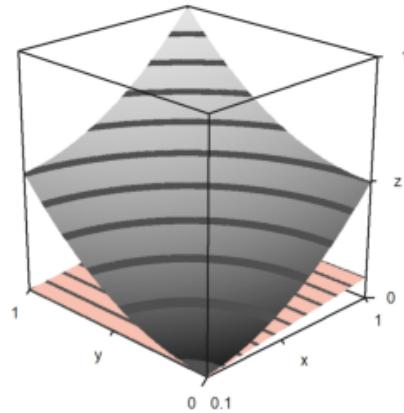
Fungsi `plot3d` bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukan program 3D, Euler dapat menggabungkan beberapa objek dasar. Kami mencoba untuk memvisualisasikan paraboloid dan garis singgung-nya.

```
>function myplot ...
y=0:0.01:1; x=(0.1:0.01:1)';
plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
    hues=0.5,>contour,color=orange);
h=holding(1);
plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
holding(h);
endfunction
```

Sekarang framedplot () menyediakan bingkai, dan menyetel tampilan.

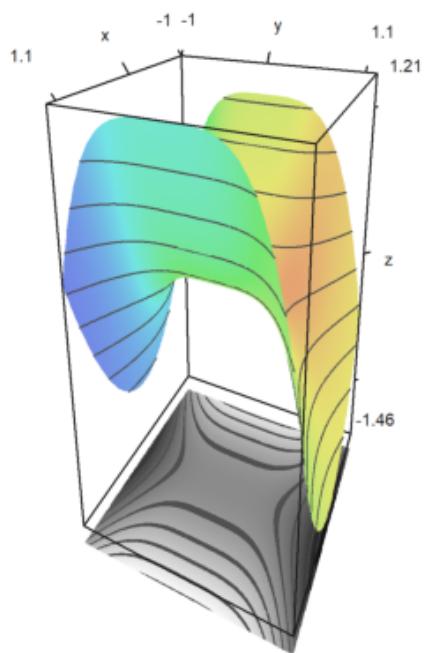
```
>framedplot("myplot", [0.1,1,0,1,0,1], angle=-45°, ...
> center=[0,0,-0.7], zoom=6):
```



Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d () menyetel jendela ke fullwindow () secara default, tetapi plotcontourplane () mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
zoom(2);
wi=fullwindow();
plotcontourplane(x,y,z,level="auto",<scale);
plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
window(wi);
reset();
endfunction

>myplot(x,y,z):
```



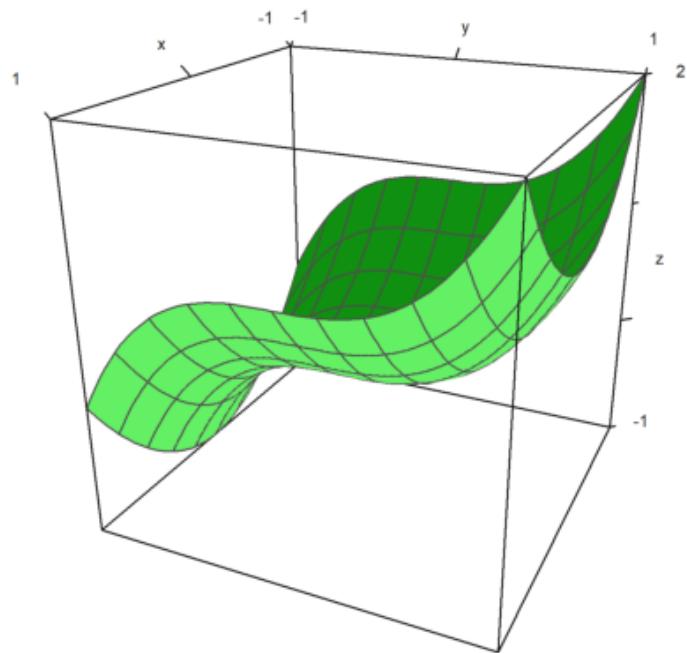
## Animasi

---

Salah satu fungsi yang memanfaatkan teknik ini adalah memutar. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi tersebut memanggil `addpage()` untuk setiap plot baru. Akhirnya itu menjawai plot.

Harap pelajari sumber rotasi untuk melihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot();
```



## Menggambar Povray

---

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan meletakkan sub-direktori "bin" Povray ke dalam jalur lingkungan, atau menyetel variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori default adalah eulerhome (), biasanya c:\Users\Username\Euler. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan file-file ini, gunakan povclear () .

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Ini dapat menghasilkan grafik fungsi  $f(x, y)$ , atau permukaan dengan koordinat X, Y, Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat pemandangan ke dalam notebook Euler.

Selain pov3d (), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, berisi kode Povray untuk objek. Untuk menggunakan fungsi ini, mulai file Povray dengan povstart (). Kemudian gunakan writeln (...) untuk menulis objek ke file adegan. Terakhir, akhiri file dengan povend (). Secara default, raytracer akan mulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi povlook () dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, Phong Shading dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk ke atas secara vertikal, sumbu a nd x, y, z di tangan kanan.

Anda perlu memuat file povray.

```
>load povray;
```

Pastikan, direktori bin Povray ada di jalurnya. Jika tidak, edit variabel berikut sehingga berisi path ke povray yang dapat dieksekusi.

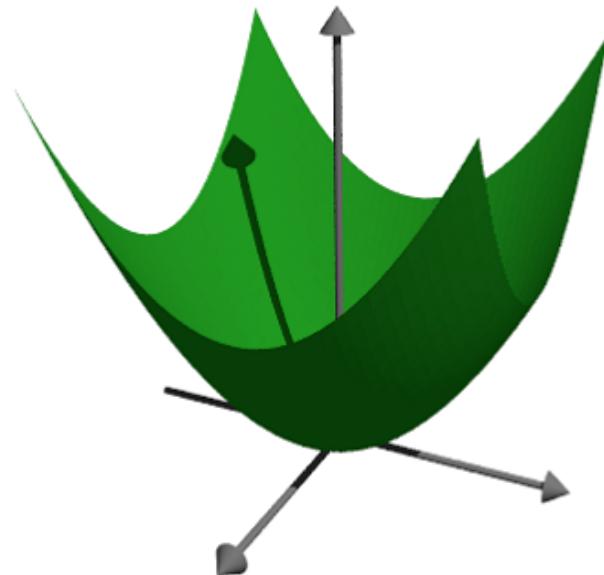
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk pertamakali, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk menelusuri file ini.

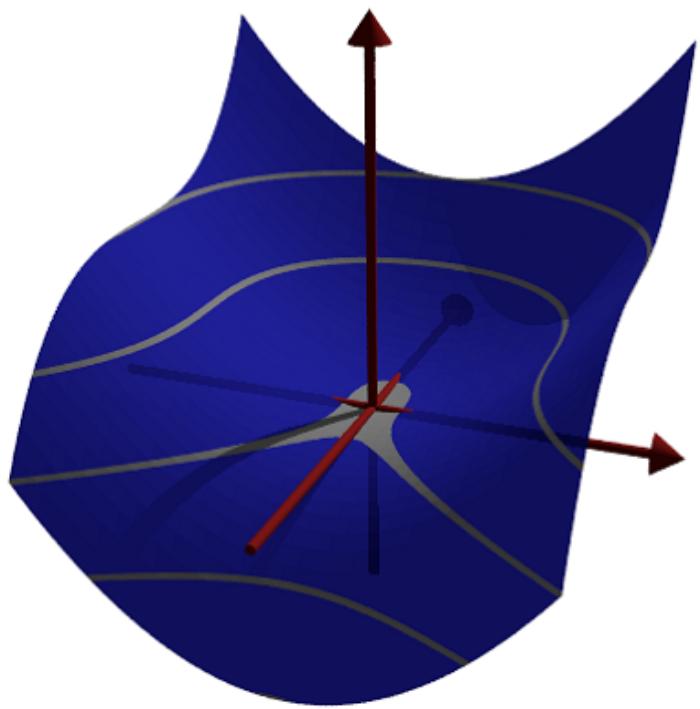
Jika Anda mulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe dijalankan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengetahui dialog start-up Povray.

```
>pov3d ("x^2+y^2", zoom=3);
```



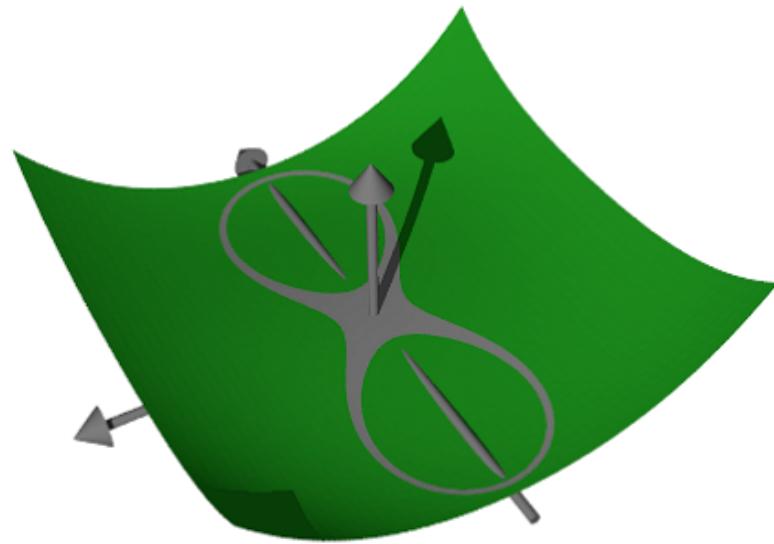
Kita bisa membuat fungsinya transparan dan menambahkan hasil akhir lainnya. Kita juga bisa menambahkan garis level ke plot fungsi.

```
>pov3d ("x^2+y^3", axiscolor=red, angle=20°, ...
>   look=povlook(blue,0.2), level=-1:0.5:1, zoom=3.8);
```



Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.  
Kami memplot himpunan titik di bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=1.5, ...
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
> <fscale,zoom=3.8);
```



## Merencanakan dengan Koordinat

Alih-alih fungsi, kita bisa memplot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objeknya.

Dalam contoh ini, kita memutar fungsi di sekitar sumbu z.

```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,8)';
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);
```



Dalam contoh berikut, kami memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...
>pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...
> w=500,h=300);
```



Dengan metode naungan lanjutan Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayangan, triknya mungkin menjadi jelas.  
Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya adalah  $[x, y, Z]$ . Kami menghitung dua turunan menjadi  $x$  dan  $y$  dari ini dan mengambil produk silang sebagai normal.

```
>dx &= diff([x, y, z], x); dy &= diff([x, y, z], y);
```

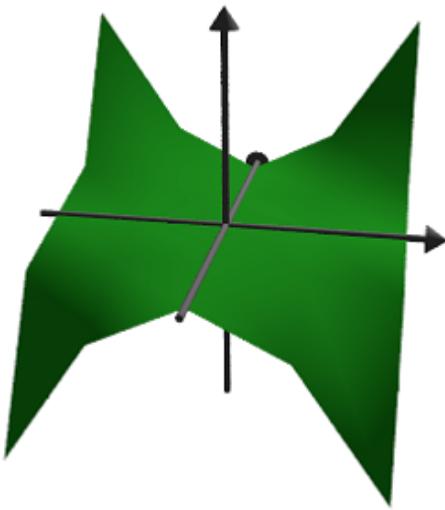
Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx, dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{matrix} 3 & 2 & 2 \\ [-2x^y, -3x^y, 1] \end{matrix}$$

We use only 25 points.

```
>x=-1:0.5:1; y=x';  
>pov3d(x, y, Z(x, y), angle=10°, ...  
> xv=NX(x, y), yv=NY(x, y), zv=NZ(x, y), <shadow>;
```



Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi perbaikannya dalam contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, tiga fungsi koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunannya menjadi x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang merupakan produk persilangan dari dua turunannya.

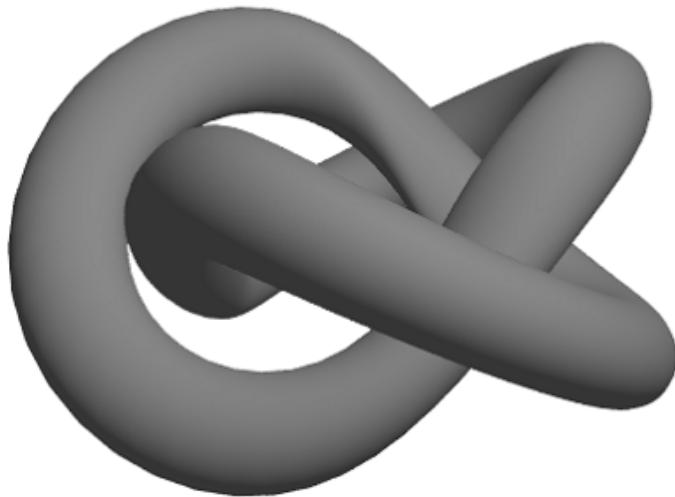
```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

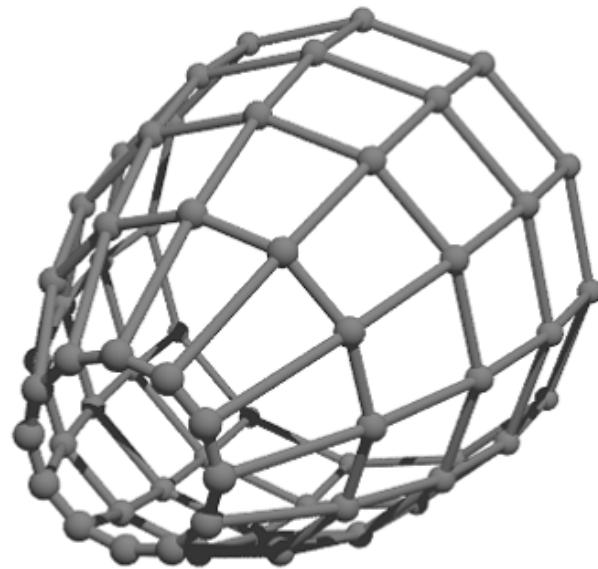
Vektor normal adalah evaluasi dari ekspresi simbolik  $dn[i]$  untuk  $i = 1, 2, 3$ . Sintaks untuk ini adalah & "expresi" (parameter). Ini adalah alternatif metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik  $NX, NY, NZ$  terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),axis=0,zoom=5,w=450,h=350, ...
>  <shadow,look=povlook(gray), ...
>  xv=&"dn[1] "(x,y), yv=&"dn[2] "(x,y), zv=&"dn[3] "(x,y));
```



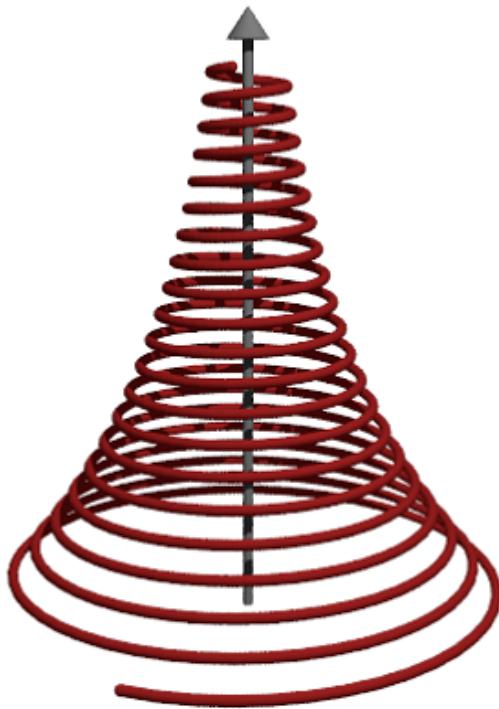
Kami juga dapat membuat grid dalam 3D.

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```



Dengan povgrid (), kurva dimungkinkan.

```
>povstart(center=[0,0,1],zoom=3.6); ...
>t=linspace(0,2,1000); r=exp(-t); ...
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
>writeln(povgrid(x,y,z,povlook(red))); ...
>writeAxis(0,2,axis=3); ...
>povend();
```



## Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray. Kami memulai output dengan povstart () .

```
>povstart (zoom=4);
```

Pertama kita tentukan tiga silinder, dan simpan dalam string di Euler.

Fungsi povx () dll. Hanya mengembalikan vektor [1,0,0], yang bisa digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(green)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c1
```

```
cylinder { <-1,0,0>, <1,0,0>, 1
texture { pigment { color rgb <0.564706,0.0627451,0.0627451> } }
finish { ambient 0.2 }
}
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Itu dilakukan oleh `povlook()`, yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kami juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

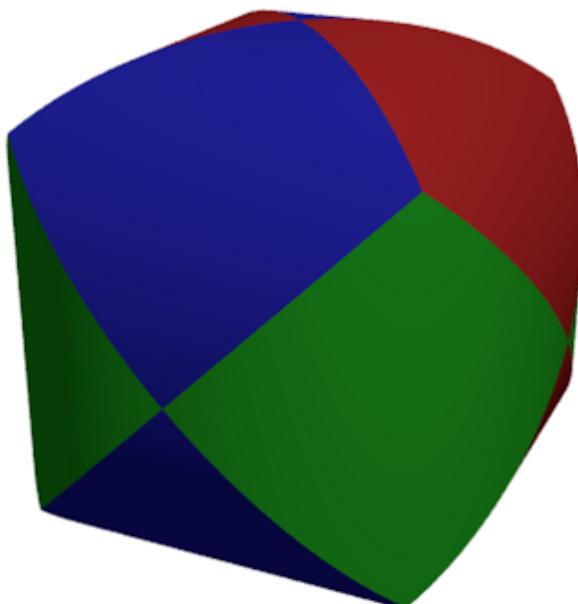
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek interseksi, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit untuk divisualisasikan, jika Anda belum pernah melihatnya sebelumnya.

```
>povend;
```



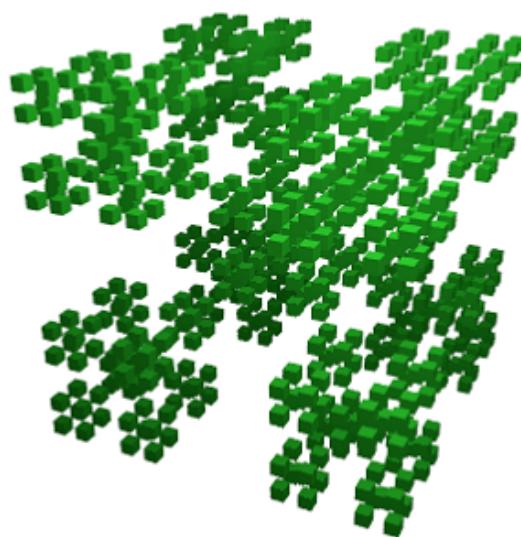
Fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi `povbox()` mengembalikan string, berisi koordinat kotak, tekstur, dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());  
>function fractal (x,y,z,h,n) ...
```

```
if n==1 then writeln(onebox(x,y,z,h));  
else  
    h=h/3;  
    fractal(x,y,z,h,n-1);  
    fractal(x+2*h,y,z,h,n-1);  
    fractal(x,y+2*h,z,h,n-1);  
    fractal(x,y,z+2*h,h,n-1);  
    fractal(x+2*h,y+2*h,z,h,n-1);  
    fractal(x+2*h,y,z+2*h,h,n-1);  
    fractal(x,y+2*h,z+2*h,h,n-1);  
    fractal(x+2*h,y+2*h,z+2*h,h,n-1);  
    fractal(x+h,y+h,z+h,h,n-1);  
endif;  
endfunction
```

```
>povstart(fade=10,<shadow);  
>fractal(-1,-1,-1,2,4);  
>povend();
```



Perbedaan memungkinkan pemotongan satu objek dari yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan sebuah objek di Povray, daripada menggunakan string di Euler. Definisi segera ditulis ke file.

Koordinat kotak -1 berarti [-1, -1, -1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita bisa menggunakan objek ini di povobject (), yang mengembalikan string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

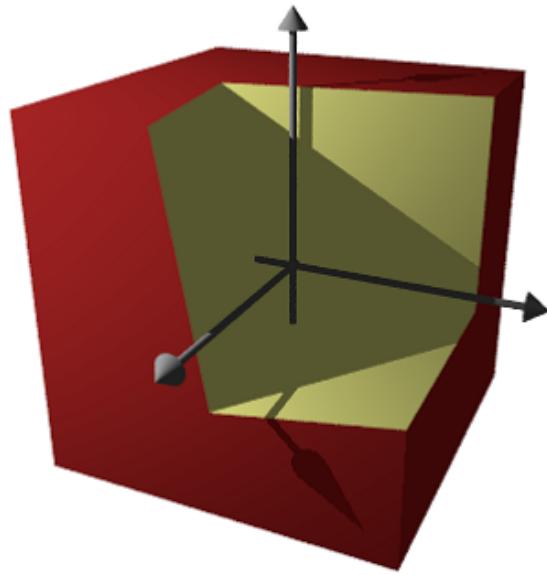
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
>    rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil perbedaan kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
>writeAxis(-1.2,1.2,axis=2); ...
>writeAxis(-1.2,1.2,axis=4); ...
>povend();
```



## Fungsi Implisit

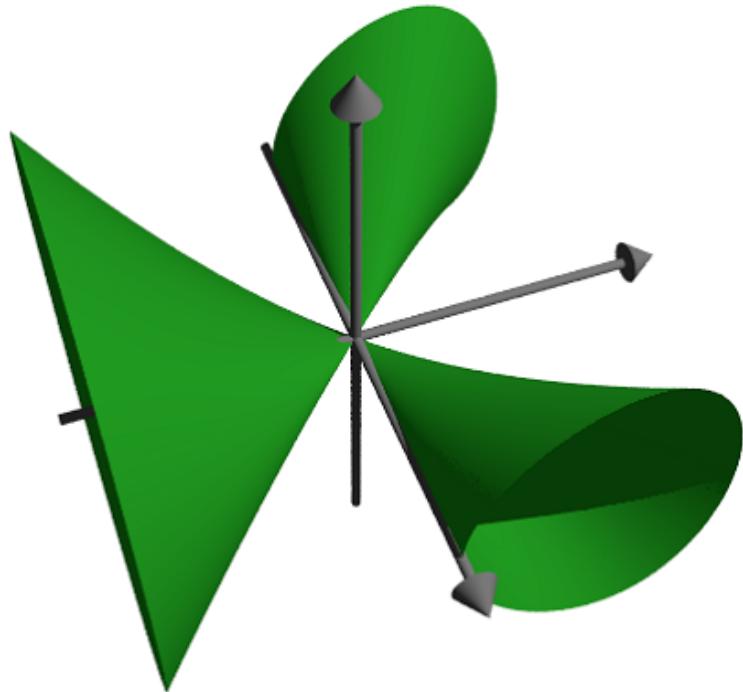
Povray dapat memplot himpunan di mana  $f(x, y, z) = 0$ , seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsinya sedikit berbeda. Anda tidak dapat menggunakan keluaran ekspresi Maxima atau Euler.

```
>povstart (angle=70°, height=50°, zoom=4);
```

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi tersebut.

```
>writeln (povsurface ("pow(x, 2) *y-pow(y, 3)-pow(z, 2)", povlook(green))); ...
>writeAxes(); ...
>povend();
```



## Objek Jaring

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi  $x + y = 1$  dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart (angle=-10°, center=[0.5, 0.5, 0.5], zoom=7);
```

Kami tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kami mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle () melakukan ini secara otomatis. Ia dapat menerima vektor normal seperti pov3d () .

Yang berikut ini mendefinisikan objek mesh, dan langsung menulisnya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kami mendefinisikan dua disk, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaan dikurangi dua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan dua persimpangan tersebut.

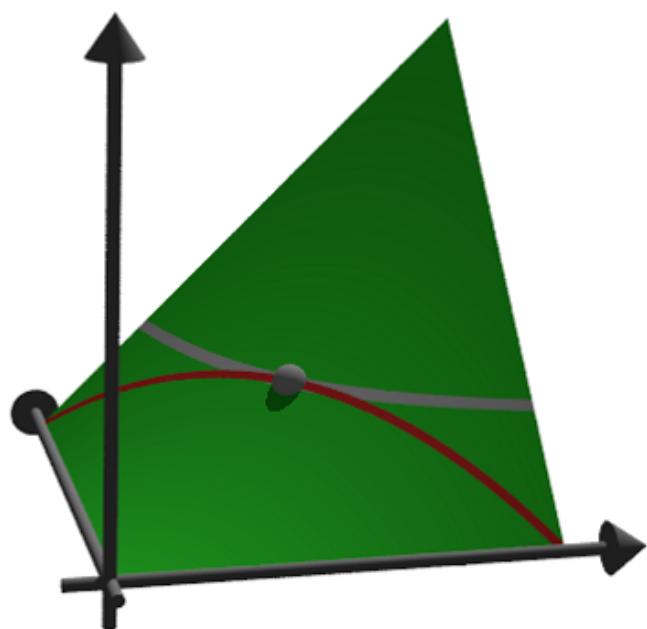
```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulis titik maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesai.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```



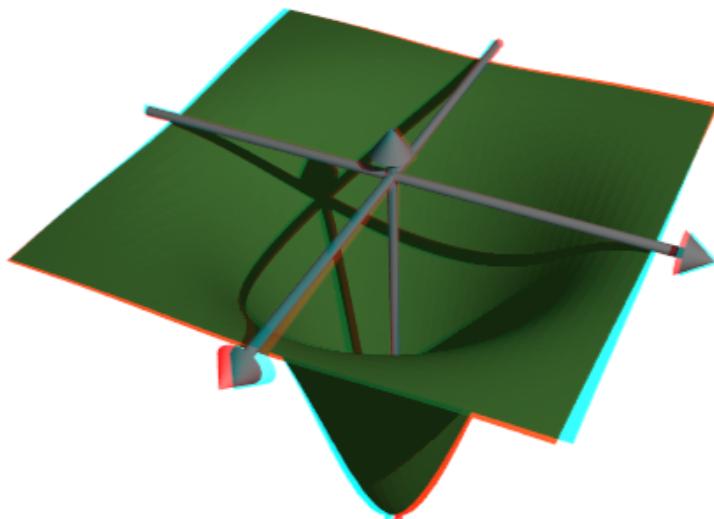
[\*\*Anaglyph di Povray\*\*](#)

Untuk menghasilkan anaglyph untuk kacamata merah / cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph () .

Tentu saja, Anda memerlukan kaca mata merah / cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d () memiliki tombol sederhana untuk menghasilkan anaglyph.

```
>pov3d "-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
> center=[0,0,0.5],zoom=3.5;
```

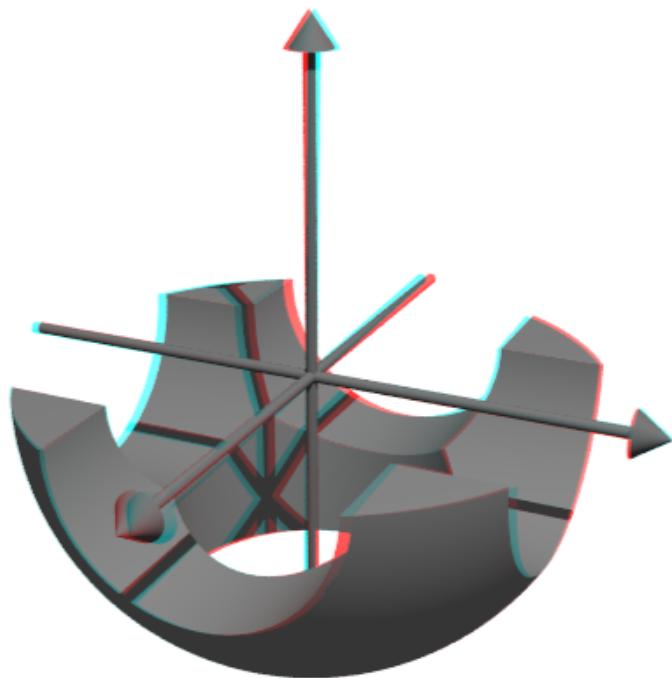


Jika Anda membuat adegan dengan objek, Anda perlu memasukkan pembuatan adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
s=povsphere(povc,1);
cl=povcylinder(-povz,povz,0.5);
clx=povobject(cl,rotate=xrotate(90°));
cly=povobject(cl,rotate=yrotate(90°));
c=povbox([-1,-1,0],1);
un=povunion([cl,clx,cly,c]);
obj=povdifference(s,un,povlook(red));
writeln(obj);
writeAxes();
endfunction
```

Fungsi povanaglyph () melakukan semua ini. Parameternya seperti di povstart () dan povend () digabungkan.

```
>povanaglyph ("myscene", zoom=4.5);
```



## Mendefinisikan Objek Sendiri

Antarmuka povray Euler berisi banyak objek. Tetapi Anda tidak dibatasi untuk ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang sama sekali baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada asalnya.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {" + r1 + "," + r2 + look + "}";
endfunction
```

Here is our first torus.

```
>t1=povdonat (0.8,0.2)
```

```
torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
    rotate 90 *x
    translate <0.8,0,0>
}
```

Sekarang kami menempatkan objek ini ke dalam sebuah adegan. Untuk tampilan, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

```
> povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, itu tidak menampilkan kesalahan. Karena itu Anda harus menggunakan

```
> povend (<exit>);
```

jika ada yang tidak berhasil. Ini akan membuat jendela Povray terbuka.

```
>povend (h=320,w=480);
```

```
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Berikut adalah contoh yang lebih lengkap. Kami menyelesaikannya

$$Ax \leq b, \quad x \geq 0, \quad c.x \rightarrow \text{Max.}$$

dan menunjukkan poin yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Yes, it has.

Next we define two objects. The first is the plane

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
endfunction
```

Kemudian kami mendefinisikan perpotongan dari semua setengah spasi dan sebuah kubus.

```
>function adm (A, b, r, look "") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
endfunction
```

We can now plot the scene.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

The following is a circle around the optimum.

-Terjemahan

Berikut ini adalah lingkaran di sekitar optimal.

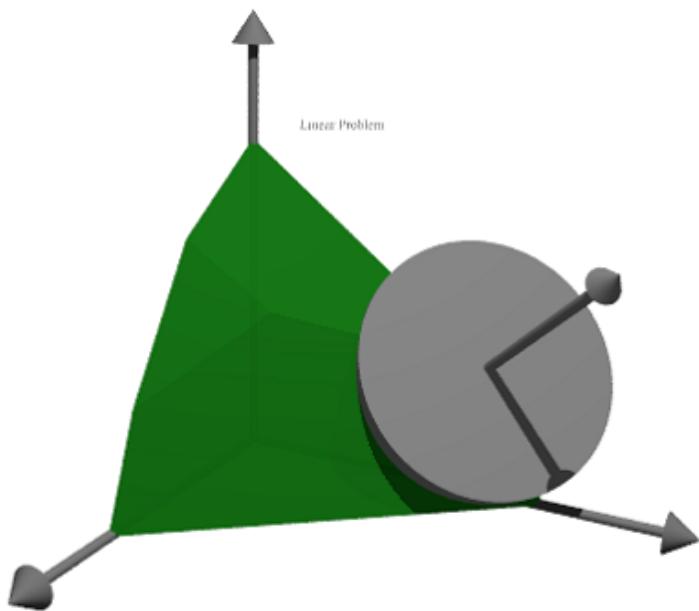
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>  povlook(red,0.9)));
```

Dan ada kesalahan di arah optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem", [0,0.2,1.3],size=0.05,rotate=125°)); ...
>povend();
```

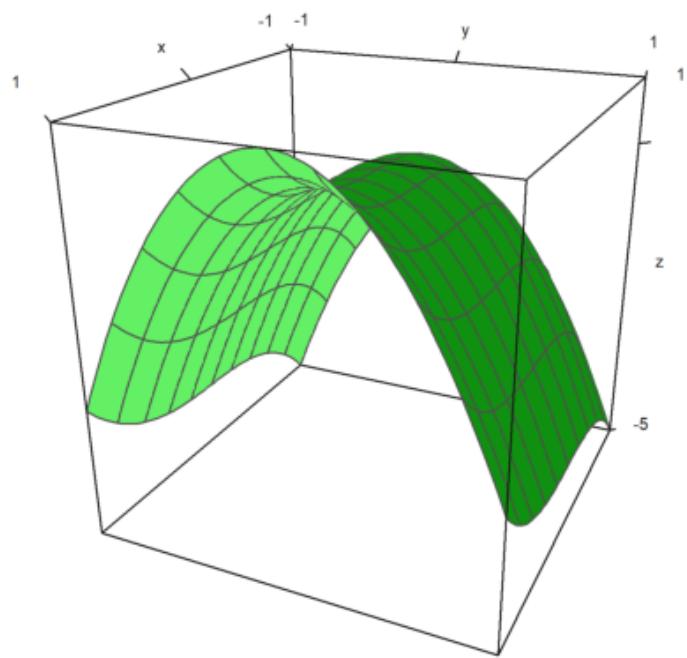


## Latihan Soal

- 
1. Buatkan grafik

$$f(x) = x^3 - 4y^2$$

```
>plot3d("x^3-4*y^2") :
```

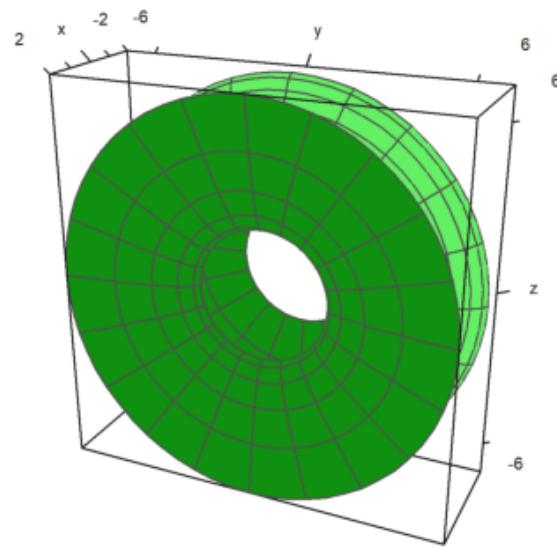


2. Buatkan grafik

$$f(x) = x^3 - 3x + 4$$

dengan  $a=-2$ ,  $b=2$ , dan  $\text{grid}=18$

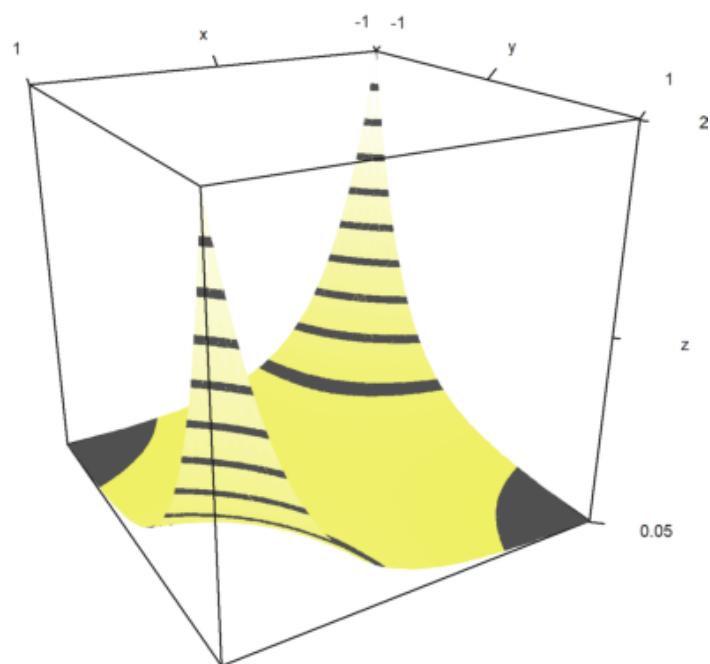
```
>plot3d("x^3-3*x+4", a=-2, b=2, rotate=true, grid=18):
```



3. Buatlah grafik plot kontur

$$f(x) = e^{3xy}$$

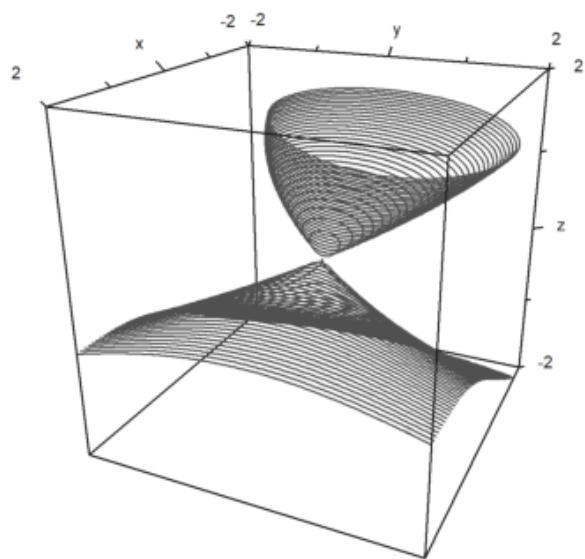
```
>plot3d("exp(3*x*y)",angle=150°,>contour,color=yellow):
```



4. Buat grafik plot implisit

$$x^2 + y^2 + 5xz + 3z^3$$

```
>plot3d("x^2+y^2+5*x*z+3*z^3",>implicit,r=2,zoom=2.2):
```



---

---

## BAB 5

---

# KB PEKAN 9-10: MENGGUNAKAN EMT UNTUK KALKULUS

### Kalkulus dengan EMT

---

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi) - Limit Fungsi, - Turunan Fungsi, - Integral Tak Tentu, - Integral Tentu dan Aplikasinya, - Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak). **Mendefinisikan Fungsi**

---

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format  $nama_fungsi := rumus_fungsi(untuk\ fungsinumerik)$ , - Menggunakan format  $nama_fungsi = rumus_fungsi(untuk\ fungsisimbolik)$ , namun dapat dihitung secara numerik, - Menggunakan format  $nama_fungsi = rumus_fungsi(untuk\ fungsisimbolikmurni)$ , tidak dapat dihitung langsung), - Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi.

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

1  
4.31977682472  
20.7392088022

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

0

```
>g(0)
```

0

```
>g(1)
```

```
Floating point error!
Error in sqrt
Error in return result.
g: useglobal; return sqrt(x^2-3*x)/(x+1)

Error in :
g(1)
^
```

Nb: Floating point error karena untuk  $x=1$ ,  $g(x)$  akan bernilai imajiner yaitu

$$\frac{\sqrt{-2}}{2}$$

```
>f(g(5)) // komposisi fungsi
```

2.20920171961

```
>g(f(5))
```

0.950898070639

```
>f(0:10) // nilai-nilai f(1), f(2), ..., f(10)
```

[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,  
99.929, 130.69, 163.51, 200.58]

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,  
99.929, 130.69, 163.51, 200.58]

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format `:=`, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

```
>function map f(x) ...
    if x>0 then return x^3
    else return x^2
    endif;
endfunction
```

```
>f(1)
```

```
1
```

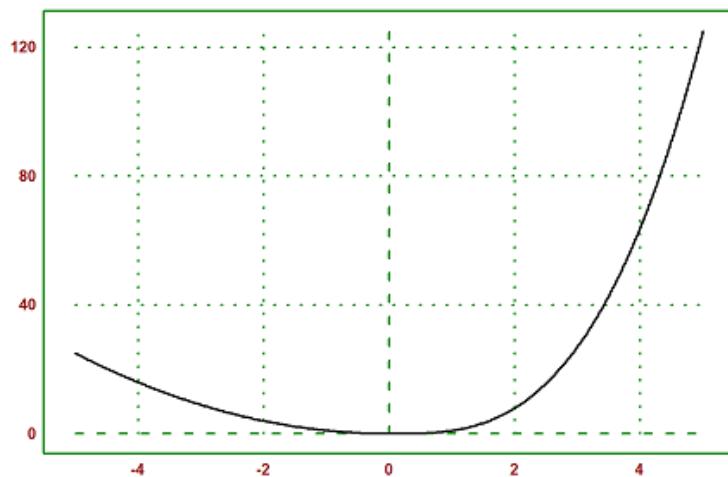
```
>f(-2)
```

```
4
```

```
>f(-5:5)
```

```
[25, 16, 9, 4, 1, 0, 1, 8, 27, 64, 125]
```

```
>aspect(1.5); plot2d("f(x)", -5, 5):
```



```
>function f(x) &= 2*x^2 // fungsi simbolik
```

$$2x^2$$

```
>function g(x) &= 3*x+1
```

$$3x + 1$$

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

$$\begin{matrix} 3 & x & + & 1 \\ 2 & E \end{matrix}$$

## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik tersebut.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

1. Untuk fungsi

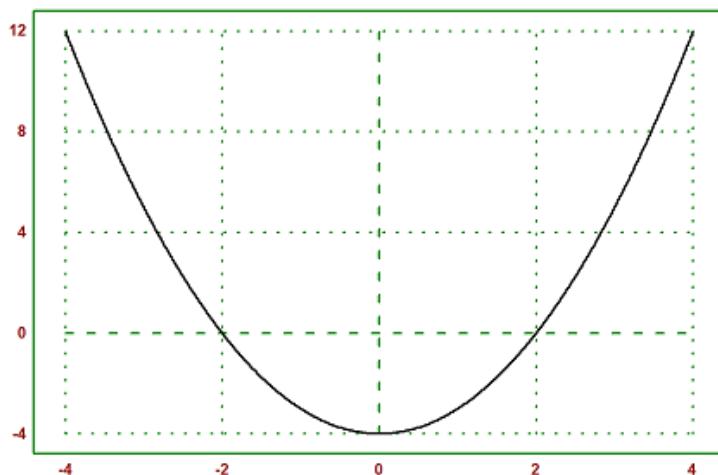
$$k(x) = x^2 - 4$$

tentukan nilai a.  $k(-4)$  b.  $k(4)$

```
>function k(x) := x^2 - 4  
>k(-4), k(4)
```

12  
12

```
>plot2d("k", -4, 4):
```



## 2. Untuk fungsi

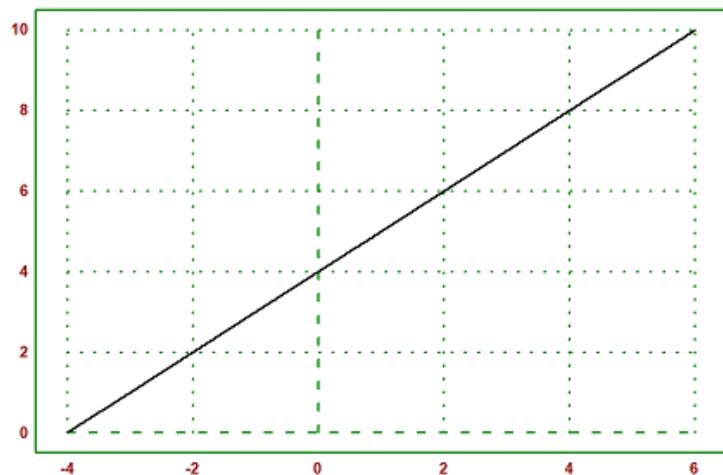
$$z(x) = \frac{x^2 - 16}{x - 4}$$

hitunglah masing-masing nilai. a.  $z(6)$  b.  $z(2)$

```
>function z(x) := (x^2-16) / (x-4)
>z(6), z(2)
```

10  
6

```
>plot2d("z", -4, 6) :
```



## 3. Untuk fungsi

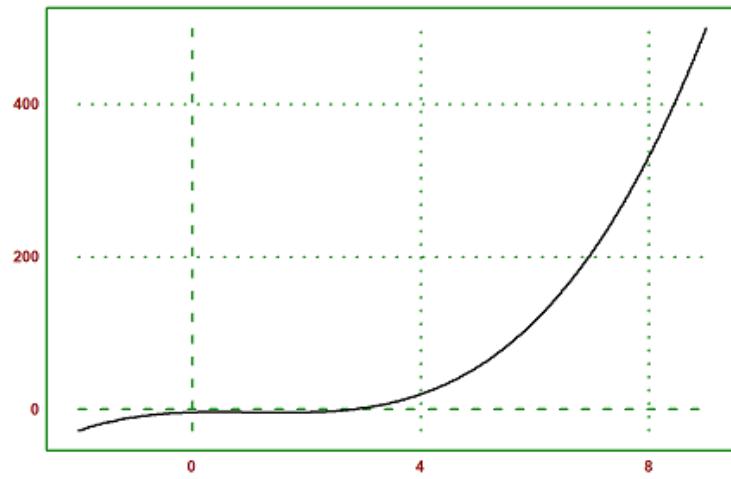
$$r(x) = x^3 - 3x^2 + 2x - 4$$

tentukan nilai  $r(4)$ ,  $r(-6)$ ,  $r(8)$

```
>function f(x) := x^3-3*x^2+2*x-4
>f(4), f(-6), f(8)
```

20  
-340  
332

```
>plot2d("x^3-3*x^2+2*x-4", -2, 9) :
```



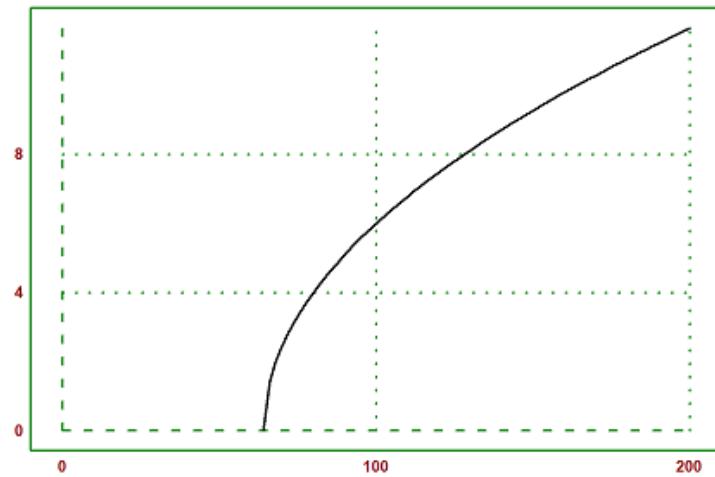
4. Tentukan nilai  $f(200)$  dari fungsi berikut

$$f(x) = \sqrt{x - 64}$$

```
>function f(x) := sqrt(x-64)
>f(200)
```

11.6619037897

```
>plot2d("sqrt(x-64)", 0, 200):
```



5. Untuk fungsi

$$f(x) = x^2 - 3x + 2$$

dan

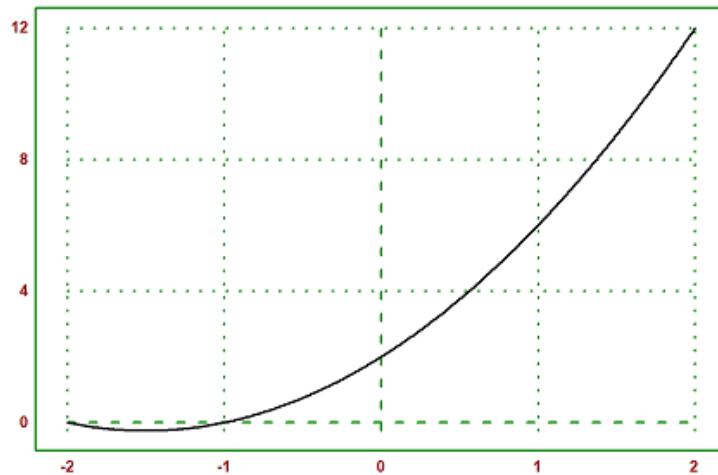
$$g(x) = x + 3$$

cari nilai  $f \circ g(-4)$ ,  $g(f(0))$

```
>function f(x) := x^2-3*x+2; $f(x)
>function g(x) := x+3; $g(x)
>f(g(-4)), g(f(0))
```

6  
5

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



6. Tentukan nilai dari

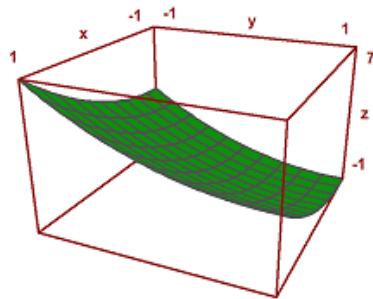
$$f(x, y) := x^2 + y^2 + 2x - 2y + 1$$

dengan  $x=1$  dan  $y=3$

```
>function f(x,y) := x^2+y^2+2*x-2*y+1
>f(1,3)
```

7

```
>plot3d("x^2+y^2+2*x-2*y+1"):
```



## Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga). Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x \rightarrow 0} \frac{1}{2x - 1} = -1$$

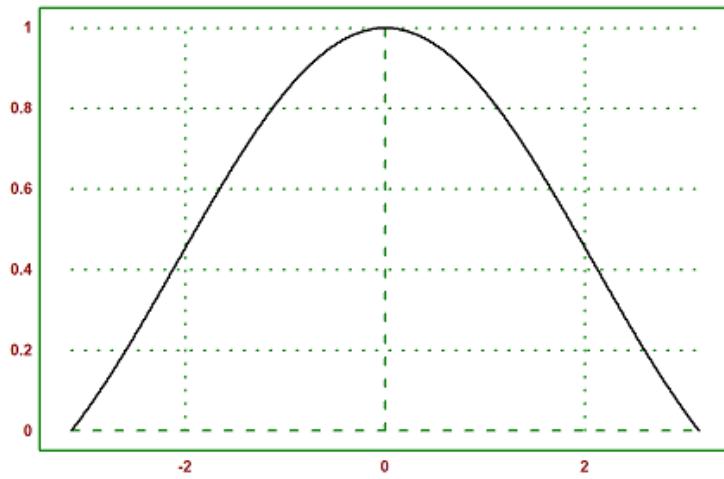
```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi):
```



```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

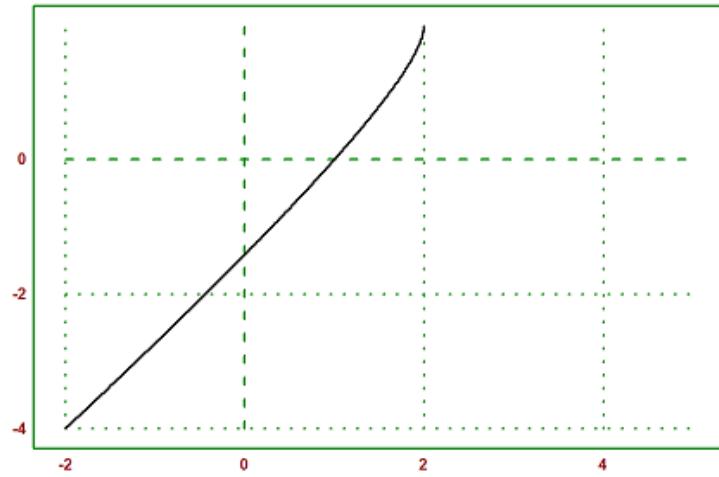
```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3}i$$

```
>plot2d("x-sqrt(2-x)",-2,5):
```



```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{1}{x} + 1 \right)^x = e$$

```
>$showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{k}{x} + 1 \right)^x = e^k$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x \rightarrow 0} (x+1)^{\frac{1}{x}} = e$$

```
>showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{x}{x+k} \right)^x = e^{-k}$$

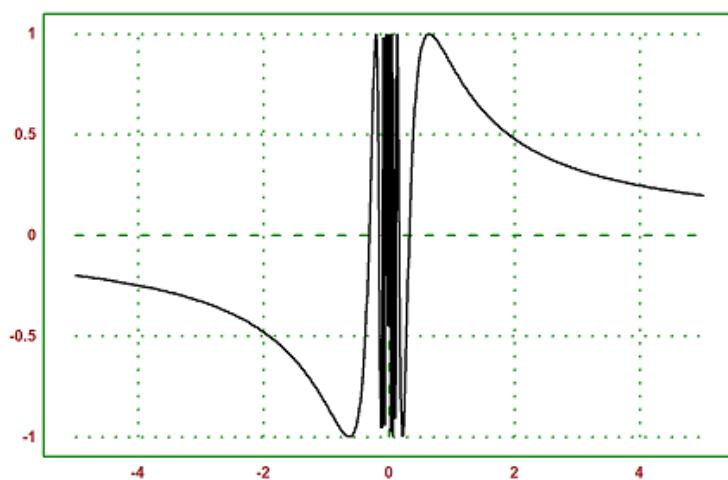
```
>showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)", -5, 5):
```



## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

1. Hitunglah nilai limit berikut.

$$\lim_{x \rightarrow 3} (x - 8)$$

```
>showev('limit((x-8),x,3))
```

$$\lim_{x \rightarrow 3} x - 8 = -5$$

2. Hitunglah nilai limit berikut.

$$\lim_{x \rightarrow 2} \frac{x^2 - 4}{x + 2}$$

```
>showev('limit((x^2-4)/(x-2),x,2))
```

$$\lim_{x \rightarrow 2} \frac{x^2 - 4}{x} = \frac{x^2 - 4}{2} = (0 = 0)$$

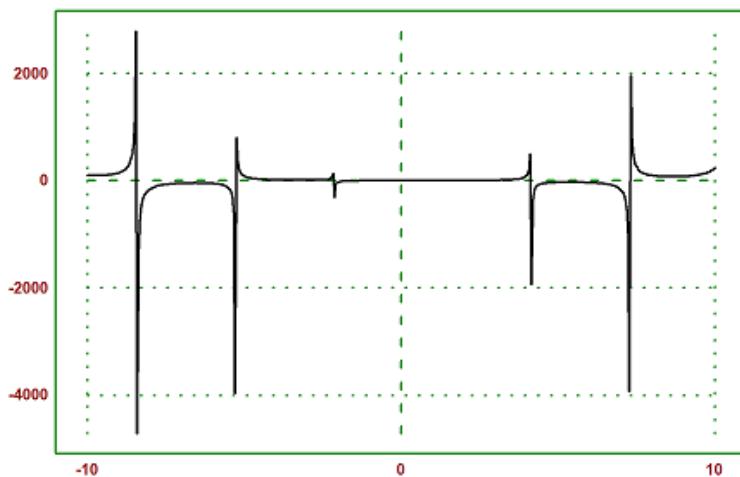
3. Hitunglah nilai limit berikut dan gambarlah grafiknya.

$$\lim_{t \rightarrow 1} \frac{t^2 - 1}{\sin(t-1)}$$

```
>showev('limit((t^2-1)/sin(t-1),t,1))
```

$$\lim_{t \rightarrow 1} \frac{t^2 - 1}{\sin(t-1)} = 2$$

```
>(plot2d("(x^2-1)/sin(x-1)", -10, 10)):
```



4. Tentukan nilai limit berikut.

$$\lim_{x \rightarrow -1} \frac{\sqrt{1-2x}}{(4x+2)^2}$$

```
>showev('limit((sqrt(1-2*x))/((4*x+2)^2), x, -1))
```

$$\lim_{x \rightarrow -1} \frac{\sqrt{1-2x}}{(4x+2)^2} = \frac{\sqrt{3}}{4}$$

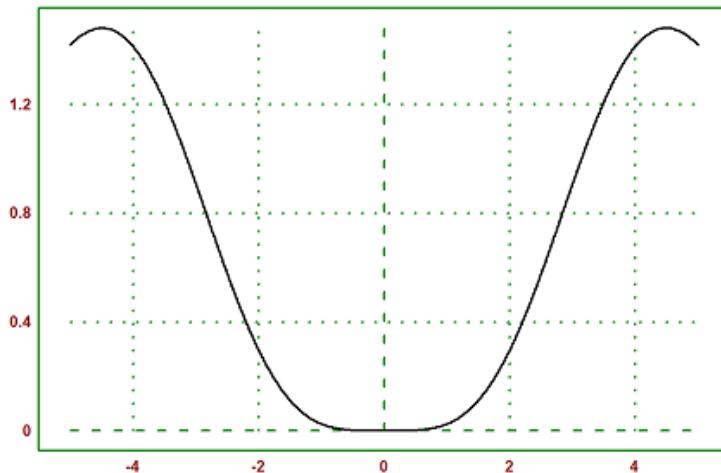
5. Tentukan nilai limit berikut.

$$\lim_{t \rightarrow 0} \frac{(t - \sin(t))^2}{t^2}$$

```
>showev('limit(((t-sin(t))^2)/(t^2), t, 0))
```

$$\lim_{t \rightarrow 0} \frac{(t - \sin t)^2}{t^2} = 0$$

```
>(plot2d("((x-sin(x))^2)/(x^2)", -5, 5) :
```



```
>
```

## Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>showev('limit(((x+h)^n-x^n)/h, h, 0)) // turunan x^n
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $(x+h)^n$  dengan menggunakan teorema binomial.

**BUKTI**

---

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Untuk

$$f(x) = x^n$$

$$\frac{d}{dx} \sin(x) = \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h}$$

Dengan

$$(a+b)^n = \sum_{k=0}^n a^k b^{n-k}$$

maka

$$\begin{aligned} &= \lim_{h \rightarrow 0} \frac{\left( x^n + \frac{n}{1!} x^{n-1} h + \frac{n(n-1)}{2!} x^{n-2} h^2 + \frac{n(n-1)(n-2)}{3!} x^{n-3} h^3 + \dots \right) - x^n}{h} \\ &= \lim_{h \rightarrow 0} \frac{n \cdot x^{n-1} h + \frac{n(n-1)}{2!} x^{n-2} h^2 + \frac{n(n-1)(n-2)}{3!} x^{n-3} h^3 + \dots}{h} \\ &= \lim_{h \rightarrow 0} n \cdot x^{n-1} + \frac{n(n-1)}{2!} \cdot x^{n-2} h + \frac{n(n-1)(n-2)}{3!} \cdot x^{n-3} h^2 + \dots \\ &= n \cdot x^{n-1} + 0 + 0 + \dots + 0 \\ &= n \cdot x^{n-1} \end{aligned}$$

Jadi, terbukti benar bahwa

$$f'(x^n) = n \cdot x^{n-1}$$

---

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $\sin(x+h)$  dengan menggunakan rumus jumlah dua sudut.

**Bukti**

---

$$f'(x) = \lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin(x)}{h}$$

$$\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$$

$$\begin{aligned} &= \lim_{h \rightarrow 0} \frac{\sin(x)\cos(h) + \cos(x)\sin(h) - \sin(x)}{h} \\ &= \lim_{h \rightarrow 0} \sin(x) \cdot \frac{\cos(h) - 1}{h} + \lim_{h \rightarrow 0} \cos(x) \cdot \frac{\sin(h)}{h} \\ &= \sin(x) \cdot 0 + \cos(x) \cdot 1 \end{aligned}$$

$$= \cos(x)$$

Jadi, terbukti benar bahwa

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


---

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = infinity$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

**Bukti**

---

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} \\ &= \lim_{h \rightarrow 0} \frac{\frac{d}{dh}(\log(x+h) - \log x)}{\frac{d}{dh}(h)} \\ &= \lim_{h \rightarrow 0} \frac{\frac{1}{x+h}}{1} \\ &= \lim_{h \rightarrow 0} \frac{1}{x+h} \\ &= \frac{1}{x} \end{aligned}$$

Jadi, terbukti benar bahwa

$$f'(x) = \lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$


---

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

Answering "Is x an ?integer?" with "unknown"  
Answering "Is x an ?integer?" with "unknown"

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h} = e^x$$

Maxima bermasalah dengan limit:

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$factor(E^(x+h)-E^x)
```

$$(e^h - 1) e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left( \lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

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

$$\frac{x}{x}$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = infinity$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

[x > 0]

```
>&forget(x<0)
```

[x < 0]

```
>&facts()
```

[ ]

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x+h) - \arcsin x}{h} = \frac{1}{\sqrt{1-x^2}}$$

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x+h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

$$\sinh(x)$$

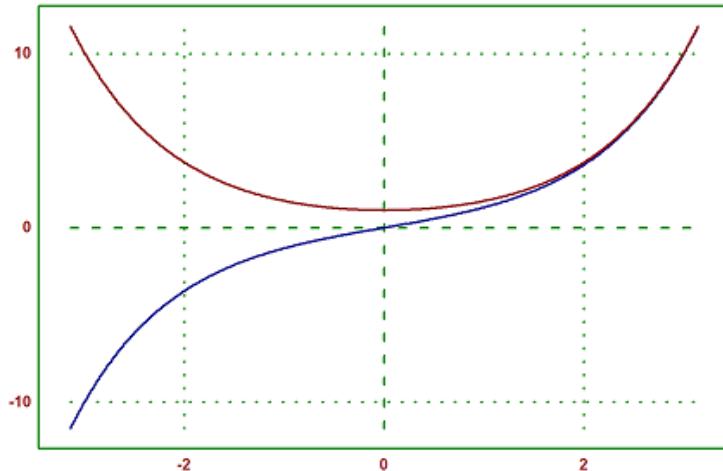
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{e^{-x} (e^{2x} + 1)}{2}$$

Hasilnya adalah  $\cosh(x)$ , karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)", "df(x)", -pi, pi, color=[blue, red]):
```



## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), seperti contoh-contoh tersebut. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

1. Tentukan nilai turunan berikut dan sketsakan grafiknya.

$$f(x) = 3x^2 + 4$$

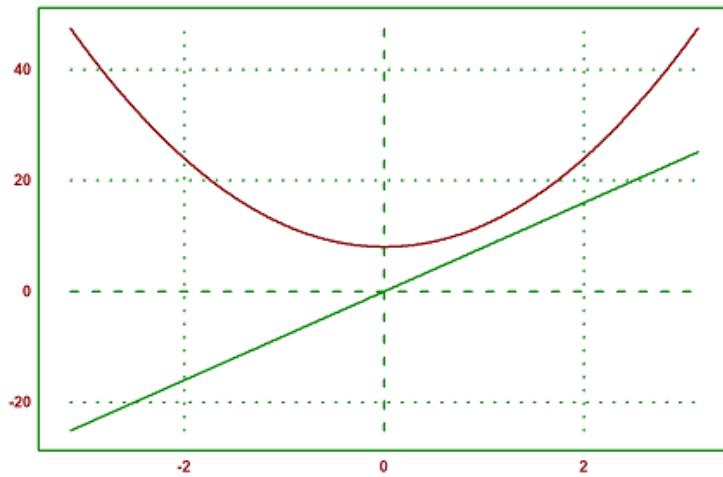
```
>function f(x) &= 4*x^2+8; $f(x)
```

$$4x^2 + 8$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); &df(x)//df(x)=f'(x)
```

8 x

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[red, green]):
```



2. Carilah turunan dari fungsi berikut

$$f(x) = \frac{4x-1}{x-2}$$

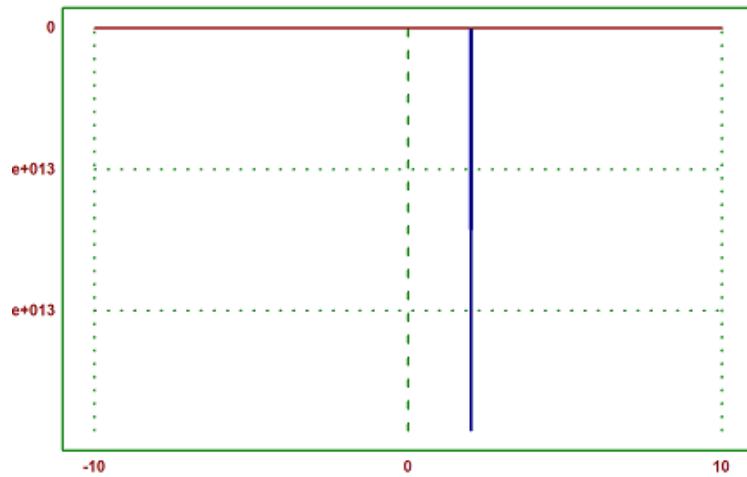
```
>function f(x) &= (x-1)/(x-2); $f(x)
```

$$\frac{x-1}{x-2}$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

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

```
>plot2d(["f(x)", "df(x)"], -10, 10, color=[blue, red]):
```



3. Carilah turunan dari fungsi berikut

$$f(x) = \frac{3}{\sqrt{x-2}}$$

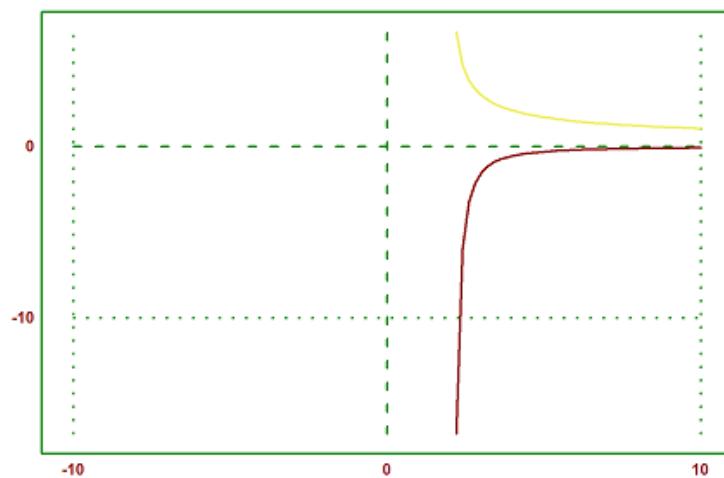
```
>function f(x) &= 3/sqrt(x-2); $f(x)
```

$$\frac{3}{\sqrt{x-2}}$$

```
>function df(x) &= limit((f(x+h)-f(x))/h, h, 0); $df(x) // df(x) = f'(x)
```

$$-\frac{3}{2(x-2)^{\frac{3}{2}}}$$

```
>plot2d(["f(x)", "df(x)"], -10, 10, color=[yellow, red]):
```



4. Carilah turunan fungsi berikut.

$$f(x) = 2\sin(x) + 3\cos(x)$$

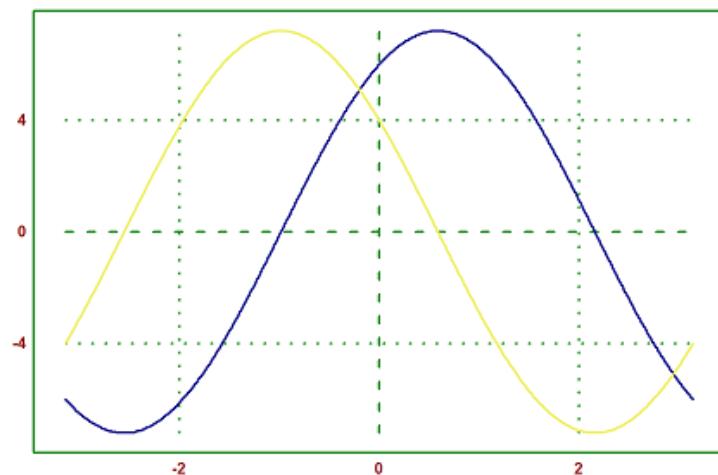
```
>function f(x) &= (4*sin(x)+6*cos(x)); $f(x)
```

$$4 \sin x + 6 \cos x$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); &df(x)
```

$$- 2 (3 \sin(x) - 2 \cos(x))$$

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, yellow]):
```



5. Tentukan turunan dan grafik fungsi berikut.

$$f(x) = \frac{\sin(x) + \cos(x)}{\cos(x)}$$

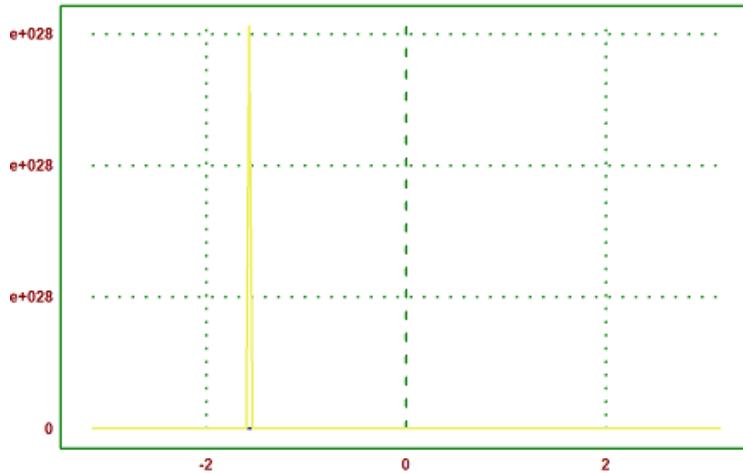
```
>function f(x) &= (\sin(x)+\cos(x)) / (\cos(x)); $f(x)
```

$$\frac{\sin x + \cos x}{\cos x}$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{\sin^2 x + \cos^2 x}{\cos^2 x}$$

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, yellow]):
```



```
>
```

## Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x) dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n, x))
```

Answering "Is n+1 zero or nonzero?" with "nonzero"

$$\int x^n dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x), x))
```

$$\int \frac{1}{x+1} dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2 + 1} dx = \arctan x$$

```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^\pi \sin x dx = 2$$

```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x dx = \cos a - \cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

Answering "Is n positive, negative, or zero?" with "positive"

$$\int_a^b x^n dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{5}{2}}}{21} - \frac{2}{105}$$

```
>$ratsimp(%)
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{7}{2}} - 2}{105}$$

```
>showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) \sin a + (e^\pi + 1) \cos a$$

```
>factor(%)
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) (\sin a - \cos a)$$

```
>function map f(x) &= E^(-x^2)
```

$$\frac{E^{-x^2}}{2}$$

```
>showev('integrate(f(x),x))
```

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi  $f$  tidak memiliki antiturunan, integralnya masih memuat integral lain.

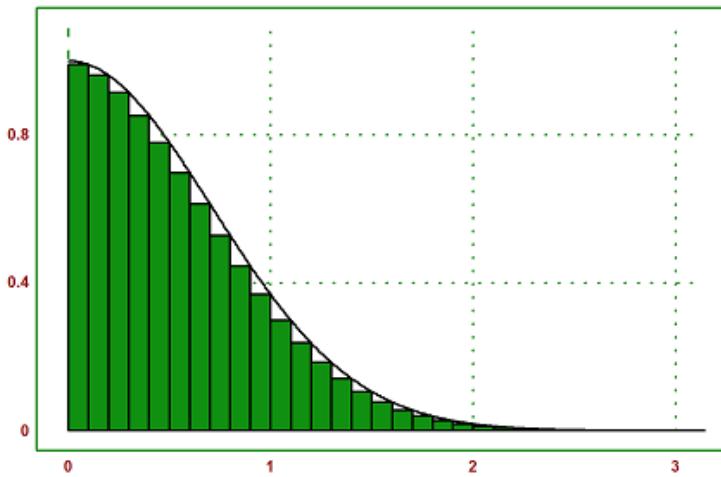
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

```
maxima: 'integrate(f(x),x,0,pi)
```

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



### Integral tentu

maxima: 'integrate(f(x),x,0,pi)

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva  $y=f(x)$  tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
>/> jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

maxima: 'integrate(f(x),x,0,pi) = 0.1\*sum(fx[i],i,1,length(fx))

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai  $f(x)$  untuk  $x = 0.1, 0.2, 0.3, \dots, 3.2$ .

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

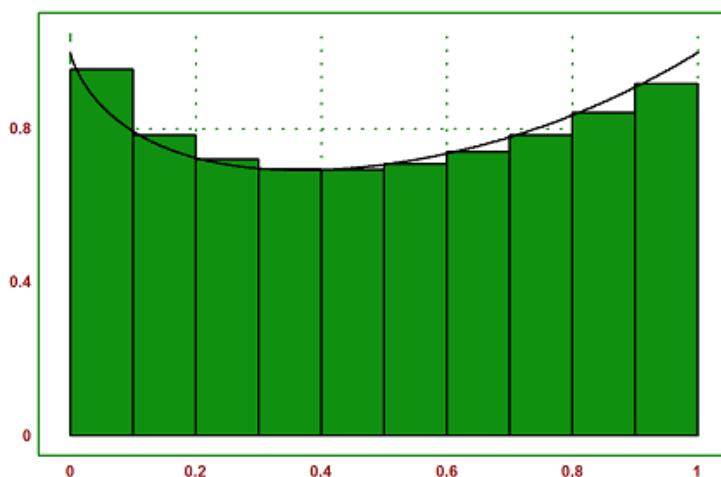
```
>function f(x) &= x^x
```

x  
x

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x \, dx = \int_0^1 x^x \, dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

maxima: 'integrate(f(x),x,0,1) = 0.01\*sum(fx[i],i,1,length(fx))

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

**Latihan**

- Bukalah buku Kalkulus. - Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). - Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut. - Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi). - Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama. - Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.) - Gunakan integral tentu untuk menghitung volume benda putar kurva  $y = f(x)$  yang diputar mengelilingi sumbu x dari  $x=a$  sampai  $x=b$ , yakni

$$V = \int_a^b \pi(f(x)^2) dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.) - Gunakan integral tentu untuk menghitung panjang kurva  $y=f(x)$  dari  $x=a$  sampai  $x=b$  dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2} dx.$$

(Pilih fungsi dan gambar kurvanya.)

1. Tentukan panjang kurva dan volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=0$  sampai  $x=4$

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

```
>function f(x) &=x^3; $f(x)
```

$$x^3$$

```
>$showev('integrate(pi*(f(x))^2,x,0,4))
```

$$\pi \int_0^4 x^6 dx = \frac{16384\pi}{7}$$

turunan fungsi  $f(x)$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$3x^2$$

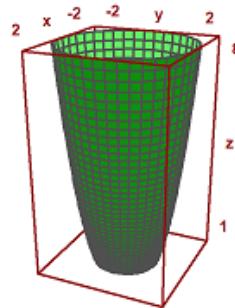
panjang kurva

```
>$showev('integrate(sqrt((1+df(x)^2)),x,0,4))
```

$$\int_0^4 \sqrt{9x^4 + 1} dx = \int_0^4 \sqrt{9x^4 + 1} dx$$

grafik benda putar mengelilingi sumbu x

```
> plot3d("x^3",2,0,2,rotate=2):
```



2. Tentukan panjang kurva dan volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=-1$  sampai  $x=2$

$$f(x) = 3x^2 + 2x + 1$$

volume benda putar

```
>function f(x) &=3*x^2+2*x+1; $f(x)
```

$$3x^2 + 2x + 1$$

```
>$showev('integrate(pi*(f(x))^2,x,-1,2))
```

$$\pi \int_{-1}^2 (3x^2 + 2x + 1)^2 dx = \frac{717\pi}{5}$$

menentukan turunan fungsi  $f(x)$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$6x + 2$$

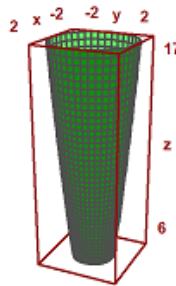
menentukan panjang kurva

```
>$showev('integrate(sqrt((1+df(x)^2)),x,-1,2))
```

$$\int_{-1}^2 \sqrt{(6x+2)^2 + 1} dx = \frac{\operatorname{asinh} 14 + 14\sqrt{197}}{12} + \frac{\operatorname{asinh} 4 + 4\sqrt{17}}{12}$$

menggambar plot benda putar mengelilingi sumbu x

```
>plot3d("3x^2+2x+1",2,-1,2,rotate=2):
```



3. Integral tentu dengan batas [-1,1] dari fungsi berikut

$$f(x) = x^2 + 8x - 9$$

```
>function map f(x) &= (x^2+8*x-9); $f(x)
```

$$x^2 + 8x - 9$$

mencari nilai dari integral tentu fungsi  $f(x)$  dengan batas [-1,1]

```
>$showev('integrate(f(x), x, -1, 1))
```

$$\int_{-1}^1 x^2 + 8x - 9 dx = -\frac{52}{3}$$

4. Tentukan panjang kurva dan volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=0$  sampai  $x=2$

$$f(x) = 4x^2 + 1$$

```
>$showev('integrate(pi*(f(x))^2,x,0,2))
```

$$\pi \int_0^2 (x^2 + 8x - 9)^2 dx = \frac{1006\pi}{15}$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$2x + 8$$

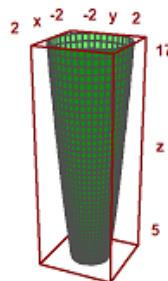
menentukan pangjang kurva

```
>$showev('integrate(sqrt((1+df(x)^2)),x,0,2))
```

$$\int_0^2 \sqrt{(2x+8)^2 + 1} dx = \frac{\operatorname{asinh} 12 + 12\sqrt{145}}{4} - \frac{\operatorname{asinh} 8 + 8\sqrt{65}}{4}$$

menggambar plot

```
>plot3d("4x^2+1",2,0,2,rotate=2):
```



5. Tentukan integral fungsi berikut.

$$f(x) = \frac{\sqrt{2x}}{x} + \frac{3}{x^5}$$

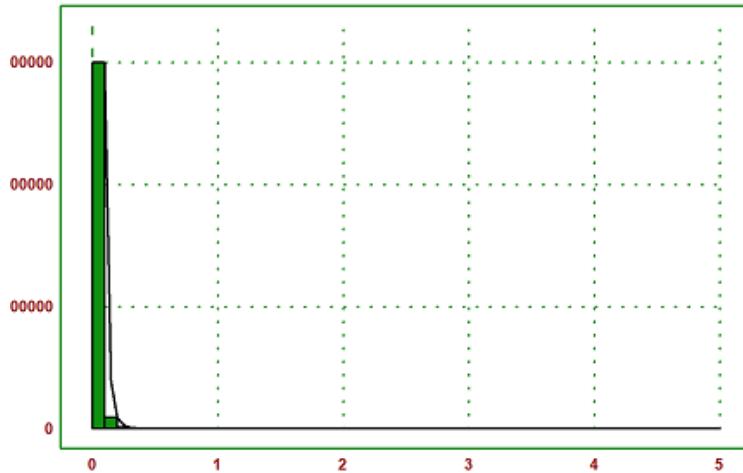
```
>function f(x) &= (sqrt(2*x))/x + 3/x^5; $f(x)
```

$$\frac{\sqrt{2}}{\sqrt{x}} + \frac{3}{x^5}$$

```
>$showev('integrate(((sqrt(2*x))/x)+(3/x^5)),x))
```

$$\int \frac{\sqrt{2}}{\sqrt{x}} + \frac{3}{x^5} dx = 2^{\frac{3}{2}} \sqrt{x} - \frac{3}{4x^4}$$

```
>x=0:0.1:5-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,5,>add):
```



## Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen berurutan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke-n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create\_list" atau "makelist" untuk menghasilkan barisan simbolik;

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
>1:2:30
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

```
>sum(1:2:30), sum(1/(1:2:30))
```

```
225  
2.33587263431
```

```
>$' sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret
```

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

```
>$' sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{3^k + k} = \sum_{k=0}^{\infty} \frac{1}{3^k + k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$' sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>$' sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$' sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{2k-1} = \sum_{k=1}^{\infty} \frac{(-1)^k}{2k-1}$$

```
>ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $' sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf), simpsum=true), &forge
```

$$a \sum_{k=0}^{\infty} x^k = \frac{a}{1-x}$$

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

```
>
```

## Deret Taylor

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar  $x=a$  adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$' e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

$$e^x = \frac{x^{10}}{3628800} + \frac{x^9}{362880} + \frac{x^8}{40320} + \frac{x^7}{5040} + \frac{x^6}{720} + \frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

```
>$' log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

$$\log x = x - \frac{(x-1)^{10}}{10} + \frac{(x-1)^9}{9} - \frac{(x-1)^8}{8} + \frac{(x-1)^7}{7} - \frac{(x-1)^6}{6} + \frac{(x-1)^5}{5} - \frac{(x-1)^4}{4} + \frac{(x-1)^3}{3} - \frac{(x-1)^2}{2} - 1$$

---

## BAB 6

---

# KB PEKAN 11-12: MENGGUNAKAN EMT UNTUK GEOMETRI

### Visualisasi dan Perhitungan Geometri dengan EMT

---

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

Numerical and symbolic geometry.

### Fungsi-fungsi Geometri

---

Fungsi-fungsi untuk Menggambar Objek Geometri:

defaultd:=textheight()\*1.5: nilai asli untuk parameter d setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r plotPoint (P, "P"): menggambar titik P dan diberi label "P" plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c" plotLabel (label, P, V, d): menuliskan label pada posisi P

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

turn(v, phi): memutar vektor v sejauh phi turnLeft(v): memutar vektor v ke kiri turnRight(v): memutar vektor v ke kanan normalize(v): normal vektor v crossProduct(v, w): hasil kali silang vektorv dan w. lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c. lineWithDirection(A,v): garis melalui A searah vektor v getLineDirection(g): vektor arah (gradien) garis g getNormal(g): vektor normal (tegak lurus) garis g getPointOnLine(g): titik pada garis g perpendicular(A, g): garis melalui A tegak lurus garis g parallel (A, g): garis melalui A sejajar garis g lineIntersection(g, h): titik potong garis g dan h projectToLine(A, g): proyeksi titik A pada garis g distance(A, B): jarak titik A dan B distanceSquared(A, B): kuadrat jarak A dan B quadrance(A, B): kuadrat jarak A dan B areaTriangle(A, B, C): luas segitiga ABC computeAngle(A, B, C): besar sudut  $\angle ABC$  angleBisector(A, B, C): garis bagi sudut  $\angle ABC$  circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r getCircleCenter(c): pusat lingkaran c getCircleRadius(c): jari-jari lingkaran c circleThrough(A,B,C): lingkaran

melalui A, B, C middlePerpendicular(A, B): titik tengah AB lineCircleIntersections(g, c): titik potong garis g dan lingkaran c circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan c2 planeThrough(A, B, C): bidang melalui titik A, B, C

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A pada sisi positif (kanan/atas) garis quad(A,B): kuadrat jarak AB spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni  $\sin(\alpha)^2 = \frac{ab}{\sqrt{(a^2 + b^2 - c^2)/2}}$  persamaan3quads dan 1spread pada segitiga dengan panjang sisi a, b, c. triplespread(sa, sb, sc) : persamaan3spreadsa, sb, sc yang menghadap sisi a, b, c. Spread Sudut Segitiga  $\text{Spread} = \sin(\phi)^2 = \frac{ab}{\sqrt{(a^2 + b^2 - c^2)/2}}$ . **Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga**

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tetapkan tiga poin dan plot mereka.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik  
>B=[0,1]; plotPoint(B,"B");  
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB  
>plotSegment(B,C,"a"); // a=BC  
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri meliputi fungsi untuk membuat garis dan lingkaran. Format garis adalah [a,b,c], yang mewakili garis dengan persamaan  $ax+by=c$ .

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1, 2, 2]
```

Hitunglah garis tegak lurus yang melalui A pada BC.

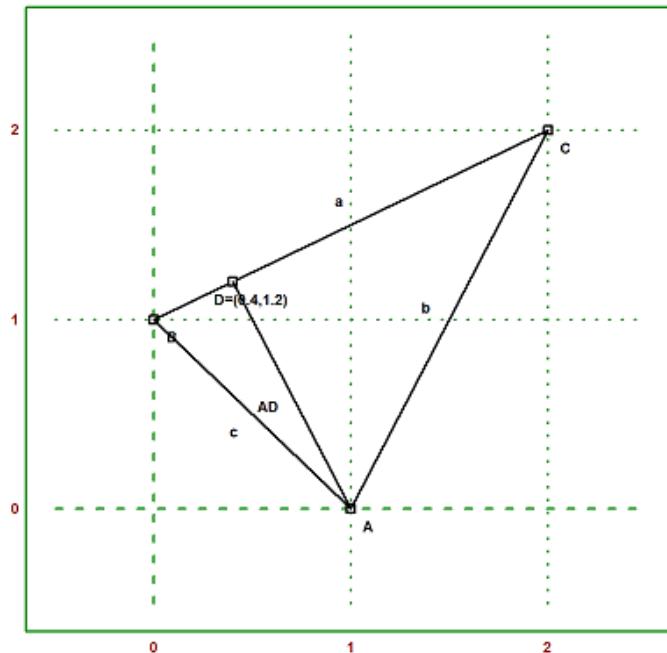
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan  
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD \cdot BC.$$

```
>norm(A-D) * norm(B-C) / 2 // AD=norm(A-D), BC=norm(B-C)
```

1.5

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D) * distance(B,C) / 2
```

1.5

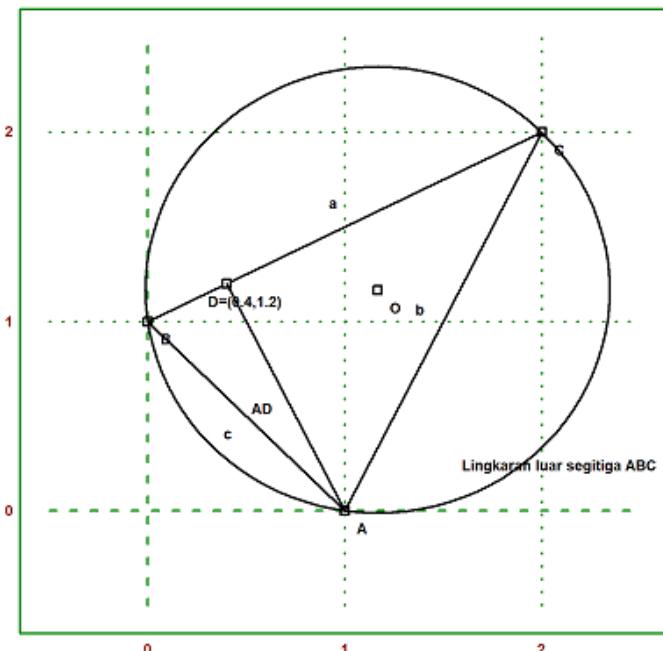
Sudut di C

```
>degrprint(computeAngle(B,C,A))
```

36°52'11.63''

Sekarang lingkaran luar segitiga.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC");
```



Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
[1.16667, 1.16667]
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

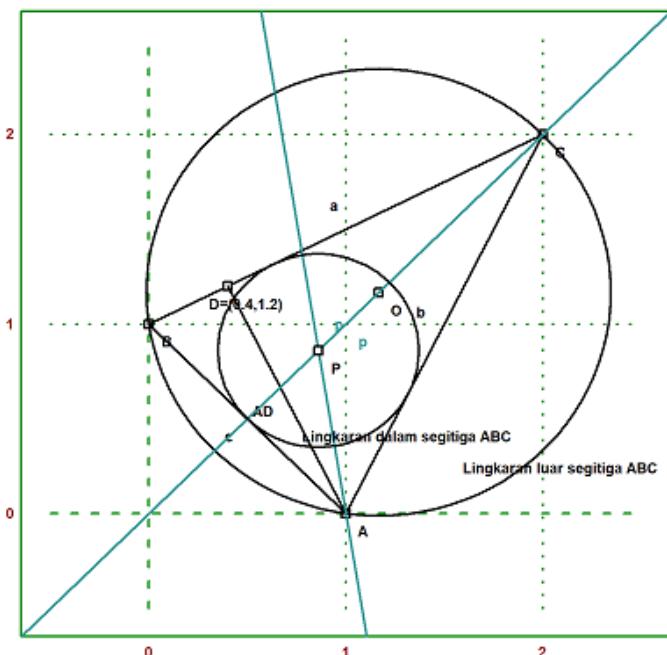
```
[0.86038, 0.86038]
```

Tambahkan semuanya ke plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut  
>plotPoint(P,"P"); // gambar titik potongnya  
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

0.509653732104

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"); // gambar lingkaran dal
```



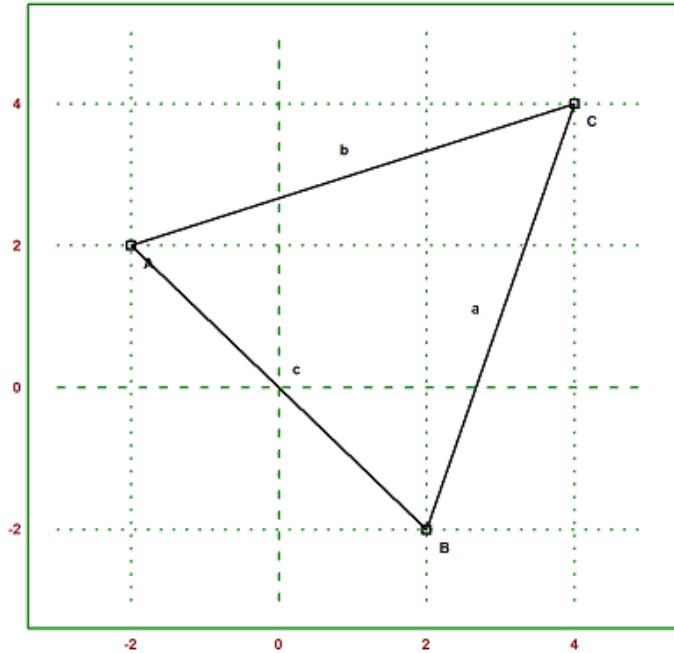
## Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(-3,5,-3,5);  
>A=[-2,2]; plotPoint(A,"A");  
>B=[2,-2]; plotPoint(B,"B");  
>C=[4,4]; plotPoint(C,"C");
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

```
>plotSegment(A,B,"c")  
>plotSegment(B,C,"a")  
>plotSegment(A,C,"b")  
>aspect(1):
```



3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

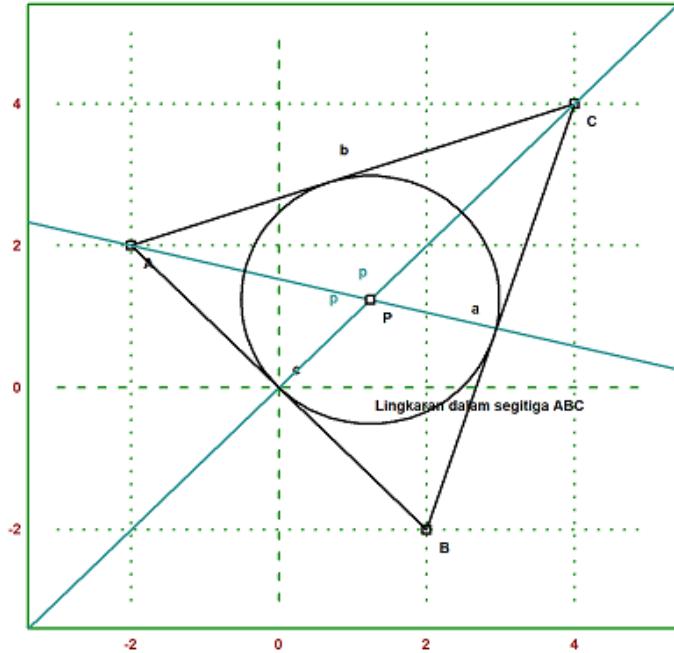
```
>l=angleBisector(A,C,B);
>g=angleBisector(C,A,B);
>P=lineIntersection(l,g)
```

```
[1.23607, 1.23607]
```

```
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

```
1.7480640978
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```



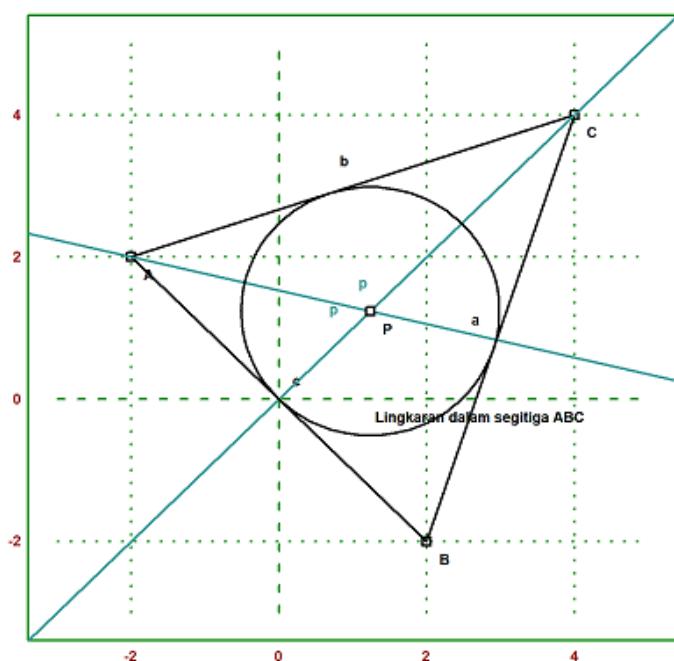
Jadi, terbukti bahwa garis bagi sudut yang ketiga juga melalui titik pusat lingkaran dalam.

#### 4. Gambar jari-jari lingkaran dalam.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

1.7480640978

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```



## Contoh 2: Geometri Simbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan perhitungan simbolis sekarang.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi memberikan perhitungan simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

```
[ - 1, 2, 2]
```

Kita bisa mendapatkan persamaan garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$2y - x = 2$$

$$\left[ y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1)y - x y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

```
[2, 1, 2]
```

```
>q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{matrix} 2 & 6 \\ [-, -] \\ 5 & 5 \end{matrix}$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[ \frac{2}{5}, \frac{6}{5} \right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[ \frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{3\sqrt{2}}$$

1.178511301977579

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 ga
```

$$\left[ \frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
[0.86038, 0.86038]
```

## Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

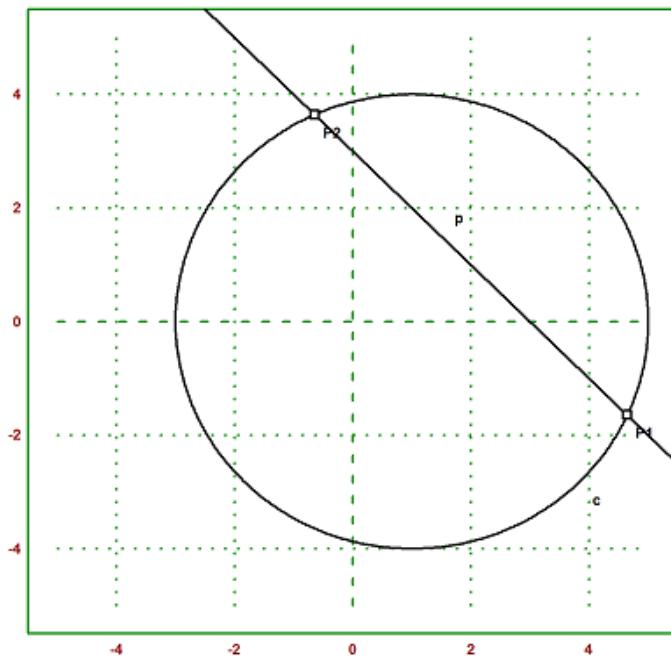
```
>A &:= [1,0]; c=circleWithCenter(A,4);  
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);  
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik potong.

```
>{P1,P2,f}=lineCircleIntersections(l,c);  
>P1, P2,
```

```
[4.64575, -1.64575]  
[-0.645751, 3.64575]
```

```
>plotPoint(P1); plotPoint(P2);
```



Begitu pula di Maxima.

```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

```
[1, 1, 3]
```

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[ \left[ \sqrt{7} + 2, 1 - \sqrt{7} \right], \left[ 2 - \sqrt{7}, \sqrt{7} + 1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

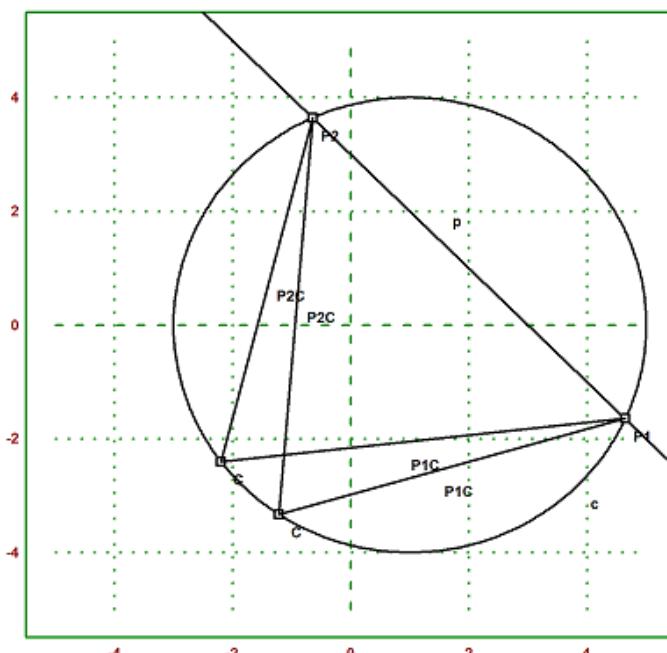
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>deprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>deprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

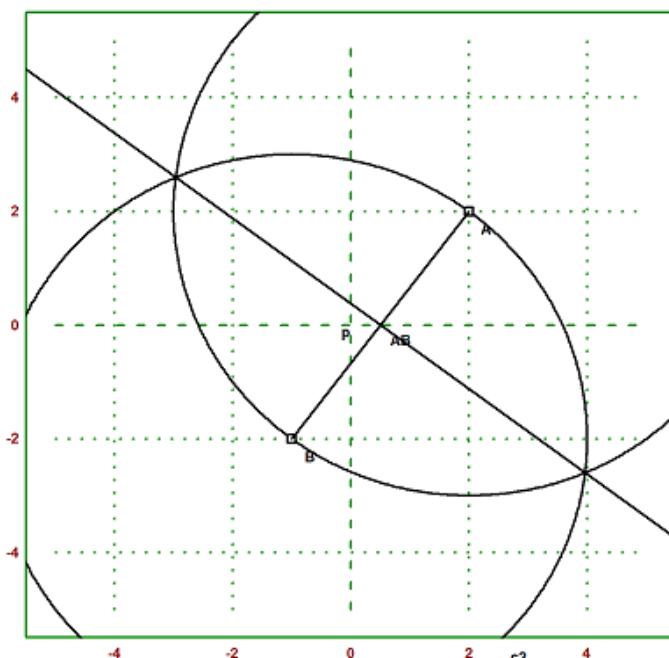
```
>insimg;
```



Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B. 2. Gambar lingkaran dengan pusat B melalui A. 3. Tarik garis melalui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l);
```



Selanjutnya, kami melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk persimpangan cukup terlibat. Tetapi kita dapat menyederhanakannya, jika kita memecahkan y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[ y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve (getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[ y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

```
>h &= getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[ y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

### Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2.$$

Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>sol &= solve ([x^2+y^2=b^2, (x-a)^2+y^2=c^2], [x,y])
```

[ ]

Ekstrak solusi y.

```
>ysol &= y with sol[2][2]; $ysol
```

Maxima said:

```
part: invalid index of list or matrix.  
-- an error. To debug this try: debugmode(true);
```

```
Error in :  
ysol &= y with sol[2][2]; $ysol  
^
```

Kami mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a, b, [1, 0, 4]) = \frac{a |ysol|}{2}$$

Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
Variable ysol not found!  
Use global or local variables defined in function H.  
Error in abs  
Error in return result.  
H: useglobal; return a*abs(ysol)/2  
  
Error in :  
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5  
^
```

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan dua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga sengan panjang sisi 3, 4, x (
```

```
Variable ysol not found!  
Use global variables or parameters for string evaluation.  
Error in abs  
Error in expression: 3*abs(ysol)/2  
%ploteval: y0=f$(x[1],args());  
adaptiveevalone: s=%ploteval(g$,t,args());  
plot2d: dw/n,dw/n^2,dw/n,auto,args());
```

Kasus umum juga berfungsi.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

Maxima said:

```
diff: second argument must be a variable; found [1,0,4]  
-- an error. To debug this try: debugmode(true);
```

```
Error in :  
$solve(diff(H(a,b,c)^2,c)=0,c)  
^
```

Sekarang mari kita cari himpunan semua titik di mana  $b+c=d$  untuk beberapa konstanta d. Diketahui bahwa ini adalah ellips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

Maxima said:

```
part: invalid index of list or matrix.  
-- an error. To debug this try: debugmode(true);
```

```
Error in :  
s1 &= subst(d-c,b,sol[2]); $s1  
^
```

Dan buat fungsi ini.

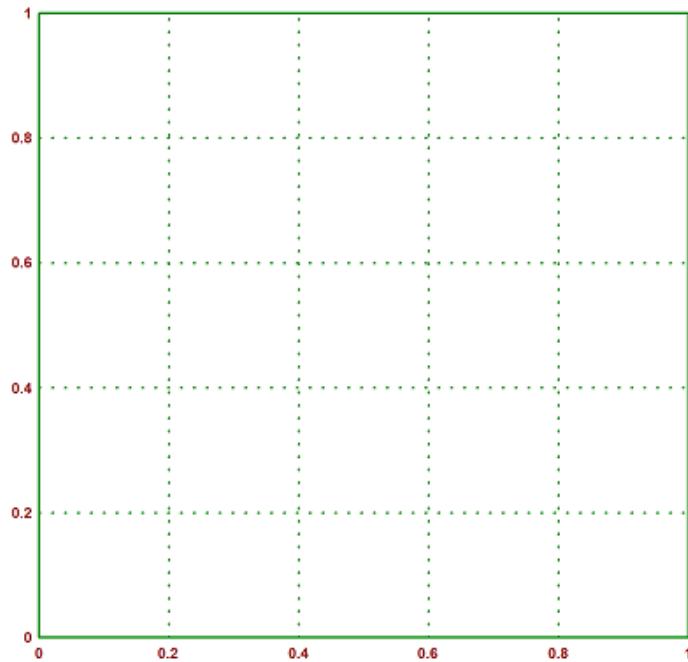
```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

0

0

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan ellips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



Kita dapat memeriksa persamaan umum untuk elips ini, yaitu.

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

di mana  $(x_m, y_m)$  adalah pusat, dan  $u$  dan  $v$  adalah setengah sumbu.

```
>ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{5 a^2}{4 u^2 - 4 u + 1}$$

Kita lihat bahwa tinggi dan luas segitiga adalah maksimal untuk  $x=0$ . Jadi luas segitiga dengan  $a+b+c=d$  maksimal jika segitiga sama sisi. Kami ingin menurunkan ini secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0, diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[ \frac{a y_{sol}^2}{2} = 0, 0 = 0 \right]$$

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusinya  $a=b=c=d/3$ .

```
>solve(eqns,[a,b])
```

$$[[a = 0, b = \%r_2]]$$

Ada juga metode Lagrange, memaksimalkan  $H(a,b,c)^2$  terhadap  $a + b + c = d$ .

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
>      diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

Maxima said:

```
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);
```

```
Error in :
diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
^
```

Kita bisa membuat plot situasinya

Pertama-tama atur poin di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

Maxima said:

```
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);
```

```
Error in :
A &= at([x,y],sol[2]); $A
^
```

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

[0,0]

[a,0]

Kemudian atur rentang plot, dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"), "B"); plotPoint(mxmeval("C"), "C"); ...
>plotPoint(mxmeval("A"), "A");
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
mxmeval:    return evaluate(mxm(s));
```

```
Error in :
plotPoint(mxmeval("A"), "A") :
^
```

Plot segmen.

```
>plotSegment(mxmeval("A"), mxmeval("C")); ...
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
mxmeval:      return evaluate(mxm(s));
```

```
Error in :
plotSegment(mxmeval("A"), mxmeval("C")); ...
^
```

```
>plotSegment(mxmeval("B"), mxmeval("C")); ...
>plotSegment(mxmeval("B"), mxmeval("A")):
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
      return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"), mxmeval("C")); plotSegment(mxmeval("B" ...
^
```

Hitung tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat lingkaran.

```
>U &= lineIntersection(h,g);
```

Kami mendapatkan rumus untuk jari-jari lingkaran.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{a_2^2 + a_1^2} \sqrt{a_2^2 + a_1^2 - 2 a a_1 + a^2}}{2 |a_2|}$$

Mari kita tambahkan ini ke plot.

```
>plotPoint(U()); ...
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2, (a2^2+a1^2-a*a1)/(2*a2)]

Error in :
plotPoint(U()); ...
^
```

```
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2, (a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
^
```

Menggunakan geometri, kami memperoleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radiusnya. Kami dapat memeriksa, apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita kuadratkan.

```
>$c^2/sin(computeAngle(A,B,C))^2 | factor
```

$$\left[ \frac{a_2^2 + a_1^2}{a_2^2}, 0, \frac{16 (a_2^2 + a_1^2)}{a_2^2} \right]$$

#### Contoh 4: Garis Euler dan Parabola

---

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

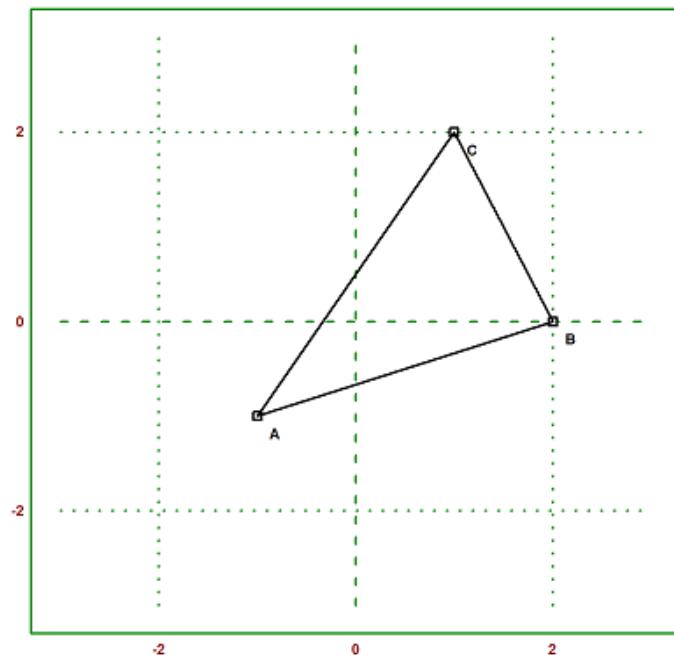
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan titik ke sana. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A, "A"); plotPoint(B, "B"); plotPoint(C, "C");
```

Kita juga bisa menambahkan sisi segitiga.

```
>plotSegment(A, B, ""); plotSegment(B, C, ""); plotSegment(C, A, "");
```



Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A, B, C)
```

$$-\frac{7}{2}$$

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A, B)
```

$$[-1, 3, -2]$$

Dan juga dapatkan rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

$$3y - x = -2$$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{3y - x + 2}{\sqrt{10}}$$
$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

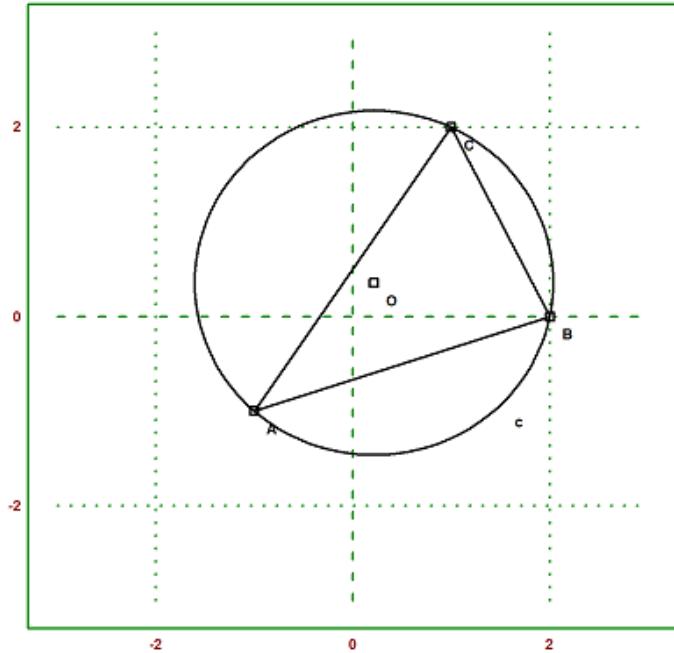
$$\left(y - \frac{5}{14}\right)^2 + \left(x - \frac{3}{14}\right)^2 = \frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

Gambarkan lingkaran dan pusatnya. Cu dan U adalah simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```



Kita dapat menghitung perpotongan ketinggian di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A, lineThrough(C, B)), ...
> perpendicular(B, lineThrough(A, C))); $H
```

$$\left[ \frac{11}{7}, \frac{2}{7} \right]$$

Sekarang kita dapat menghitung garis Euler dari segitiga.

```
>el &= lineThrough(H, O); $getLineEquation(el, x, y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kami.

```
>plotPoint(H(), "H"); plotLine(el(), "Garis Euler");
```

```
Function H needs at least 3 arguments!
Use: H (a, b, c)
```

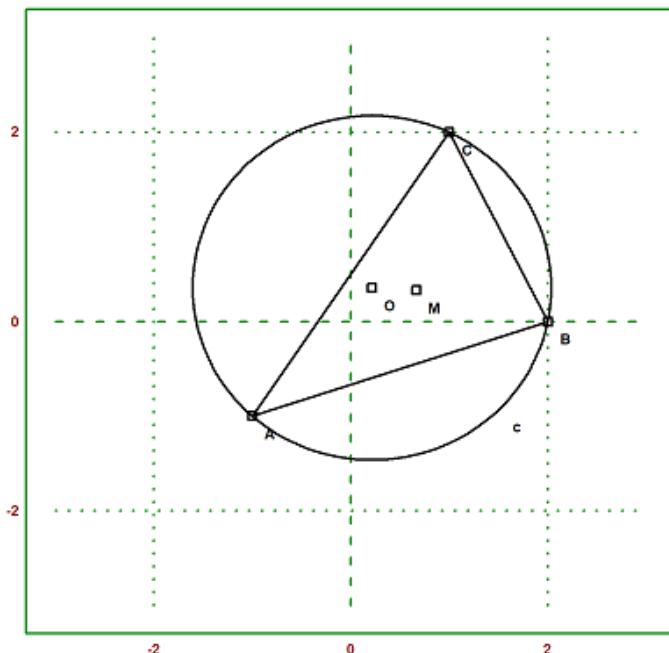
```
Error in :
plotPoint(H(), "H"); plotLine(el(), "Garis Euler");
^
```

Pusat gravitasi harus berada di garis ini.

```
>M &= (A+B+C) / 3; $getLineEquation(e1,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M"); // titik berat
```



Teorinya memberitahu kita  $MH=2*MO$ . Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
>$distance(M,H)/distance(M,O) | radcan
```

2

Fungsi termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos \left( \frac{4}{\sqrt{5} \sqrt{13}} \right)$$

$60^\circ 15' 18.43''$

Persamaan untuk pusat incircle tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

$$\left[ \frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 2^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

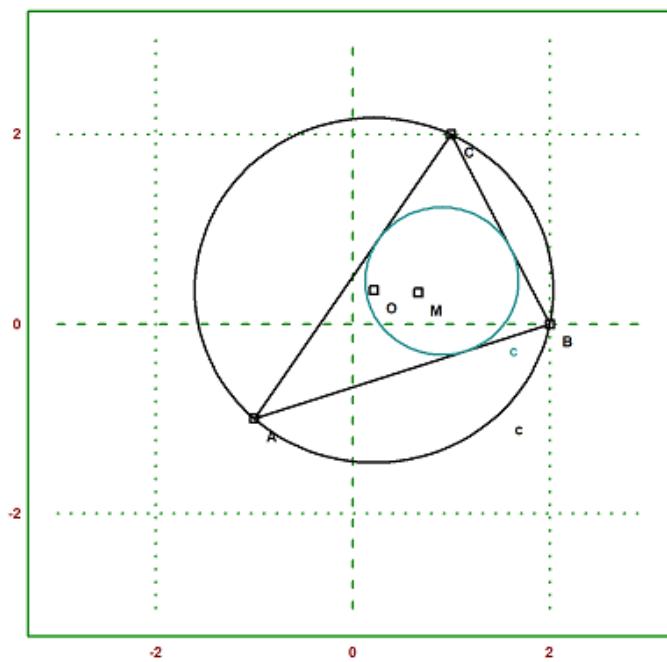
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
>color(5); plotCircle(LD()):
```



## Parabola

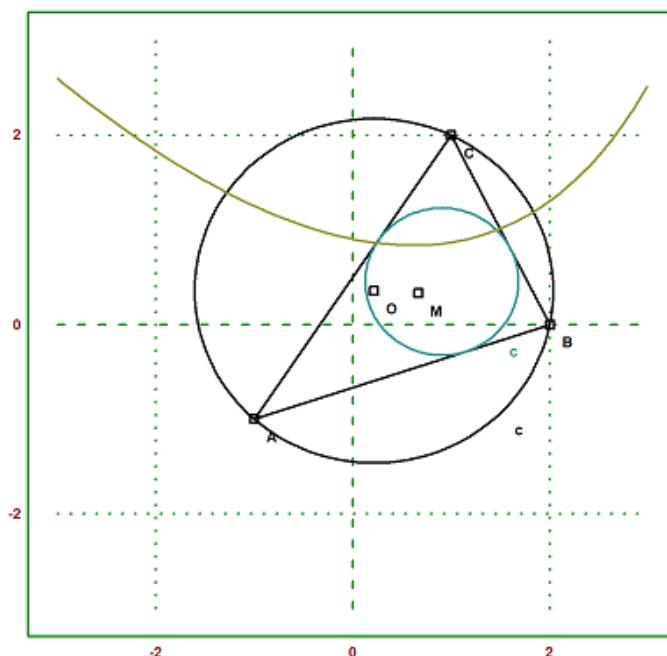
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya menjadi beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

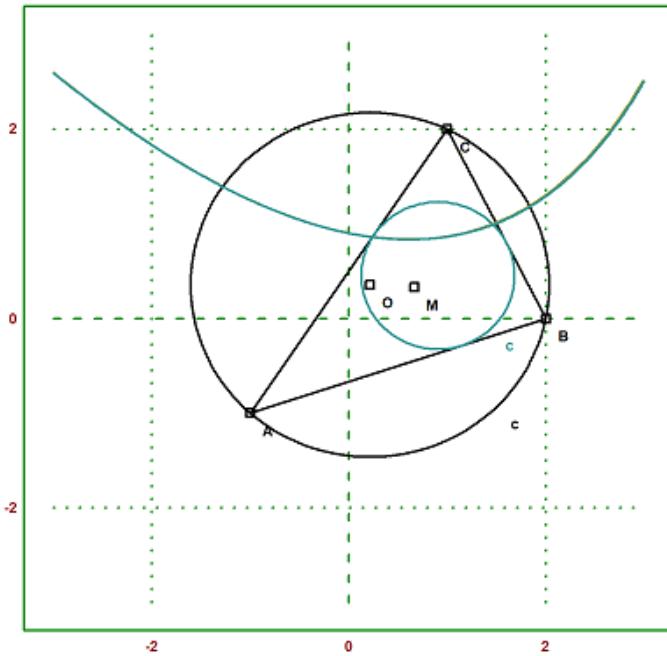
$$\begin{aligned} y &= -3x - \sqrt{70} \sqrt{9 - 2x} + 26, \\ y &= -3x + \sqrt{70} \sqrt{9 - 2x} + 26 \end{aligned}$$

Solusi pertama adalah

maxima: akar[1]

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]), add=1):
```



```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x) // fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3x - \sqrt{70}\sqrt{9-2x} + 26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

```
>U &= projectToLine(T, lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[ \frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10} \right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

2.135605779339061

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

### Contoh 5: Trigonometri

#### Rasional

---

Ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti pengertian klasik tentang jarak dan sudut dengan kuadrat dan penyebaran. Dengan menggunakan ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

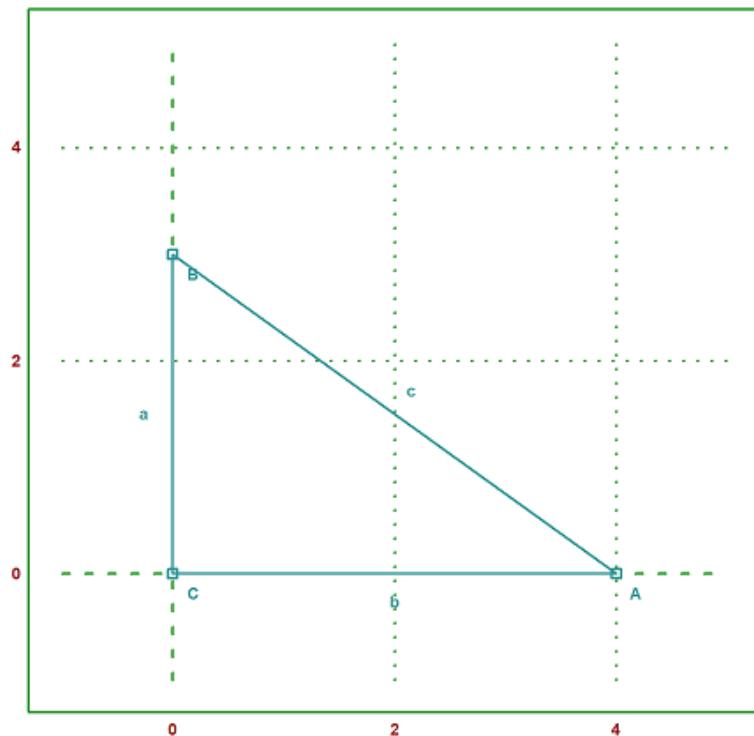
Berikut ini, saya memperkenalkan konsep, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasil tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolis sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk merencanakan geometri bidang yang terdapat dalam file Euler "geometry.e".

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana  $w_a$  adalah sudut di A. Cara yang biasa untuk menghitung sudut ini, adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

```
>wa := arcsin(3/5); degprint(wa)
```

$36^\circ 52' 11.63''$

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanya jarak kuadrat. Berikut ini, a, b, dan c menunjukkan kuadrat dari sisi-sisinya.

Teorema Pythagoras menjadi  $a+b=c$ .

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Pengertian kedua dari trigonometri rasional adalah penyebaran. Spread mengukur pembukaan antar baris. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai:

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadrat dari sembarang segitiga siku-siku dengan salah satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

9/25

Hukum kosinus trigonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Di sini a, b, dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran sudut A. Sisi a, seperti biasa, berhadapan dengan sudut A.

Hukum ini diimplementasikan dalam file geometri.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[ \left( bb - aa + \frac{7}{6} \right)^2, \left( bb - aa + \frac{7}{6} \right)^2, \left( bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[ \frac{14bb(1 - saa)}{3}, \frac{14bb(1 - saa)}{3}, \frac{52^{\frac{3}{2}}bb(1 - saa)}{3} \right]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari spread di A. Untuk melakukan ini, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk spread yang tidak diketahui sa.

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memilikinya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$1024 = 1600(1 - x)$$

$$\left[ x = \frac{9}{25} \right]$$

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum  $a, b, c$ . Hasilnya adalah rumus yang menghitung penyebaran sudut segitiga yang diberikan kuadrat dari ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[ \left[ \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36}, \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36}, \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36} \right] \right]$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e dari Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak begitu mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degsprint(arcsin(sqrt(6/7)))
```

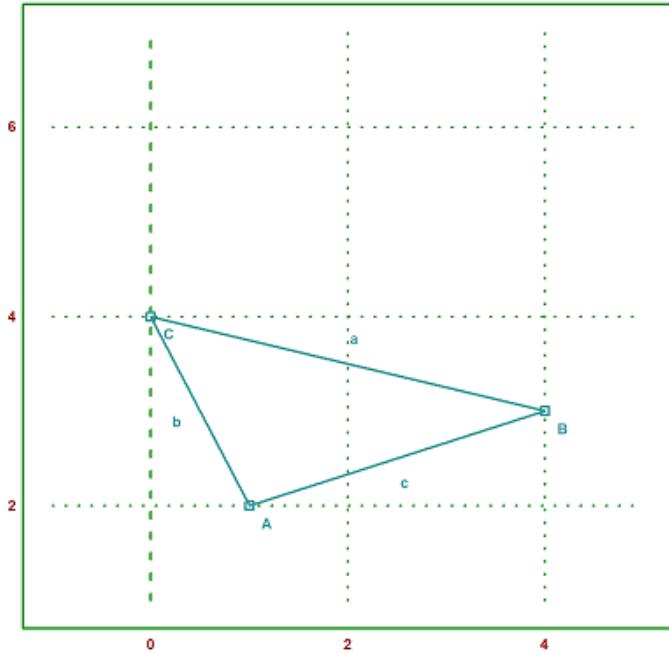
$$67^\circ 47' 32.44''$$

## Contoh lain

Sekarang, mari kita coba contoh yang lebih maju.

Kami mengatur tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
>setPlotRange(-1,5,1,7); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan jarak fungsi file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena  $c+b$  bukan  $a$ , maka segitiga itu bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$10$$

$$5$$

$$17$$

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali titik dua vektor. Hasilnya adalah beberapa pendekatan floating point.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = \mathbf{C} - \mathbf{B} = \langle -4, 1 \rangle, \quad \mathbf{c} = \mathbf{A} - \mathbf{B} = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\sqrt{17}}\right)$$

32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan x.

```
>$crosslaw(a,b,c,x), $solve(%,x),
```

$$4 = 200 (1 - x)$$

$$\left[ x = \frac{49}{50} \right]$$

Yaitu, apa yang dilakukan oleh penyebaran fungsi yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Itu menyelesaikan istilah  $\sin(\arccos(...))$  menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

Menurut definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadrat dan menyebar.

image: (20) RationalGeometryCaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadratnya.

```
>$sqrt (ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berhadapan dengan kuadrat!

```
>$sqrt (ha) *sqrt (a) / 2
```

$$\frac{7}{2}$$

Rumus determinan biasa menghasilkan hasil yang sama.

```
>$areaTriangle (B,A,C)
```

$$\frac{7}{2}$$

**Rumus Heron**

---

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue (a,b,c,sb,ha);
```

Pertama kita hitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), faktorkan dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread (b^2,c^2,a^2), $factor (%*c^2*a^2/4)
```

$$\frac{-c^4 - (-2b^2 - 2a^2)c^2 - b^4 + 2a^2b^2 - a^4}{4a^2c^2}$$
$$\frac{(-c + b + a)(c - b + a)(c + b - a)(c + b + a)}{16}$$

**Aturan Triple Spread**

---

Kerugian dari spread adalah mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread dari sebuah segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc + sb + sa)^2 = 2 (sc^2 + sb^2 + sa^2) + 4 sa sb sc$$

Aturan ini berlaku untuk setiap tiga sudut yang menambah  $180^\circ$ .

$$\alpha + \beta + \gamma = \pi$$

Sejak menyebar

$$\alpha, \pi - \alpha$$

sama, aturan triple spread juga benar, jika

$$\alpha + \beta = \gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran rangkap tiga juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung penyebaran sudut  $60^\circ$ . Ini  $3/4$ . Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[ x = \frac{3}{4}, x = 0 \right]$$

Sebaran  $90^\circ$  jelas 1. Jika dua sudut dijumlahkan menjadi  $90^\circ$ , sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan  $a,b,1$ . Dengan perhitungan berikut kita mendapatkan  $a+b=1$ .

```
>$triplespread(x,y,1), $solve(%,x)
```

$$(y + x + 1)^2 = 2 (y^2 + x^2 + 1) + 4 xy$$

$$[x = 1 - y]$$

Karena sebaran  $180^\circ-t$  sama dengan sebaran  $t$ , rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami membuat ini fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

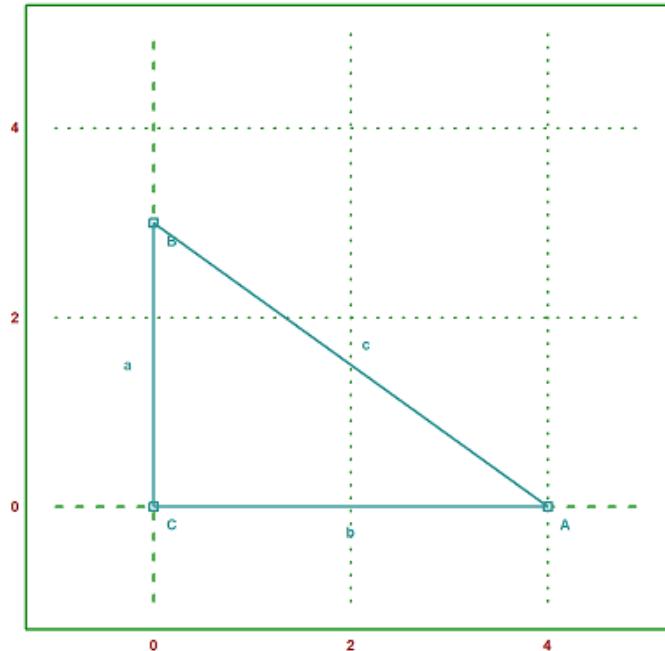
$$[x = 4a - 4a^2, x = 0]$$

$$- 4 (a - 1) a$$

## Sudut Pembagi

Ini situasinya, kita sudah tahu.

```
>C:=[0,0]; A:=[4,0]; B:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaiakannya untuk umum a,b,c.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung penyebaran sudut yang dibagi dua di A, dengan menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut terbelah  $180^\circ$ -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(% ,x), sa2 &= rhs(%[1]); $sa2
```

$$\left(2x + \frac{a}{b+a}\right)^2 = 2 \left(2x^2 + \frac{a^2}{(b+a)^2}\right) + \frac{4ax^2}{b+a}$$
$$\left[x = \frac{-\sqrt{b}\sqrt{b+a} + b + a}{2b + 2a}, x = \frac{\sqrt{b}\sqrt{b+a} + b + a}{2b + 2a}\right]$$
$$\frac{-\sqrt{b}\sqrt{b+a} + b + a}{2b + 2a}$$

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kami dapat mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arccos(sqrt(1/10)); degprint(wa2)
```

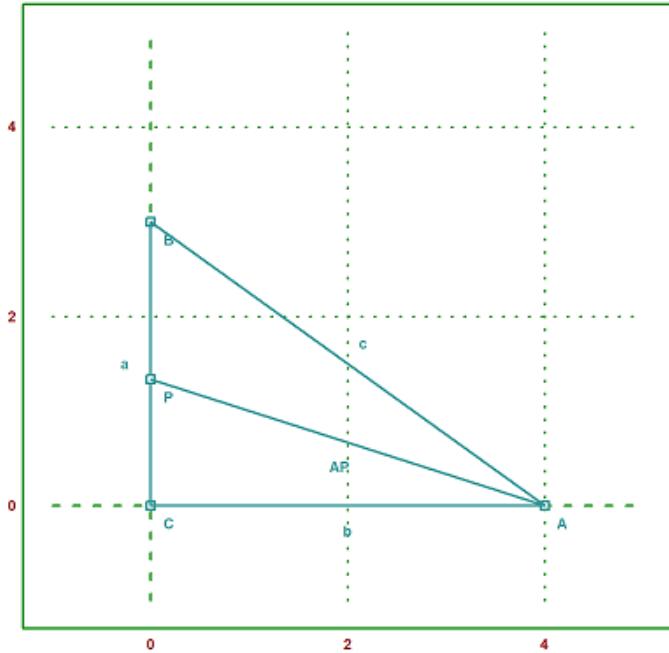
$$18^\circ 26' 5.82''$$

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

$$[0, 1.33333]$$

```
>plotPoint(P, "P"); plotSegment(A,P);
```



Mari kita periksa sudut dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku dalam segitiga apa pun. Kuadratkan, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a,b,c menunjukkan quadrances.

Karena spread CPA adalah  $1-sa^2$ , kita dapatkan darinya  $bisa/1=b/(1-sa^2)$  dan dapat menghitung bisa (kuadrant dari garis-bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356  
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus spread.

```
>py:=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b \left(\sqrt{b} \sqrt{b+a}-b-a\right)}{\sqrt{b} \sqrt{b+a}+b+a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

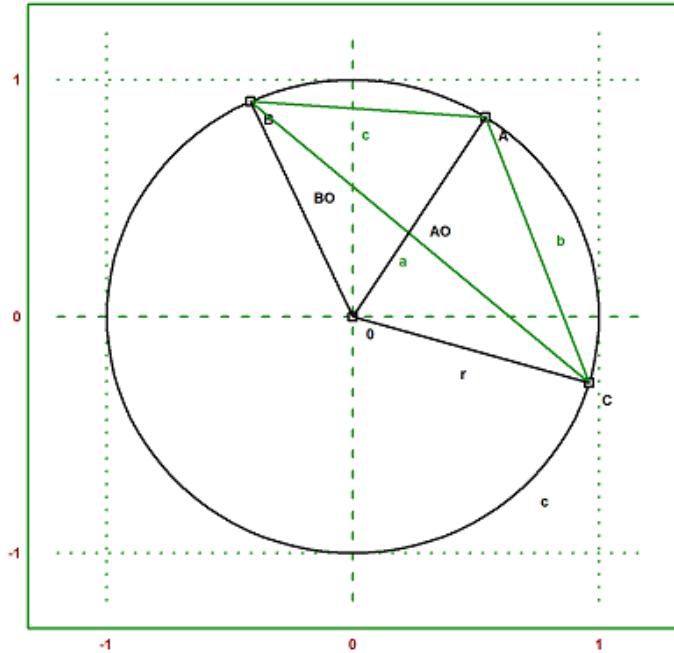
```
1.33333333333
```

## Sudut Akord

---

Perhatikan situasi berikut.

```
>setPlotRange(1.2); ...  
>color(1); plotCircle(circleWithCenter([0,0],1)); ...  
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...  
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...  
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...  
>color(1); O:=[0,0]; plotPoint(O,"O"); ...  
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...  
>insimg;
```



Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk  $r$ . Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadrat dari sisi.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

$$-\frac{abc}{c^2 - 2bc + a(-2c - 2b) + b^2 + a^2}$$

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk poin A,B,C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jarinya memang 1.

```
>periradius(a,b,c)
```

Faktanya, spread CBA hanya bergantung pada b dan c. Ini adalah teorema sudut chord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Sebenarnya spreadnya adalah  $b/(4r)$ , dan kita melihat bahwa sudut chord dari chord b adalah setengah dari sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

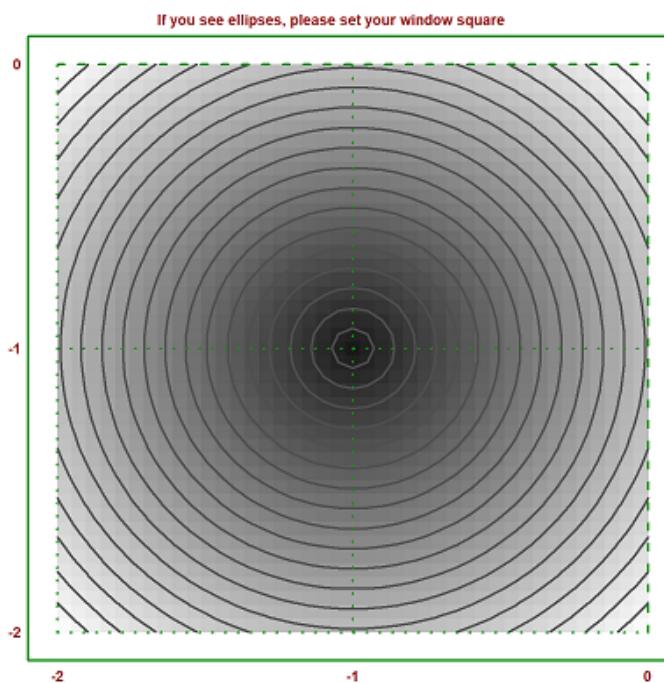
$$0$$

## Contoh 6: Jarak Minimal pada Bidang

### Catatan awal

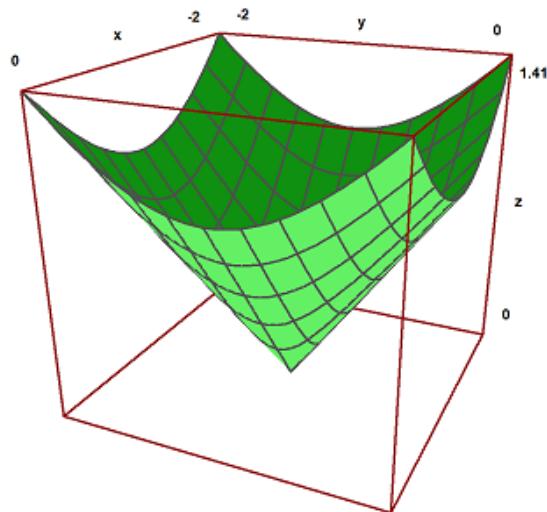
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

```
>&remvalue();  
>A=[-1,-1];  
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)  
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...  
>title="If you see ellipses, please set your window square":
```



dan grafiknya juga agak sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```



Tentu saja minimal 0 dicapai di A.

## Poin Dua

Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang diketahui" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk AB minimum yang konstan pada segmen [AB]:

```
>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```

```
Illegal function result in map.
%ploteval2:      return map(f$,x,y,args());
fcontour:    Z=%ploteval2(f$,X,Y,maps,args());
```

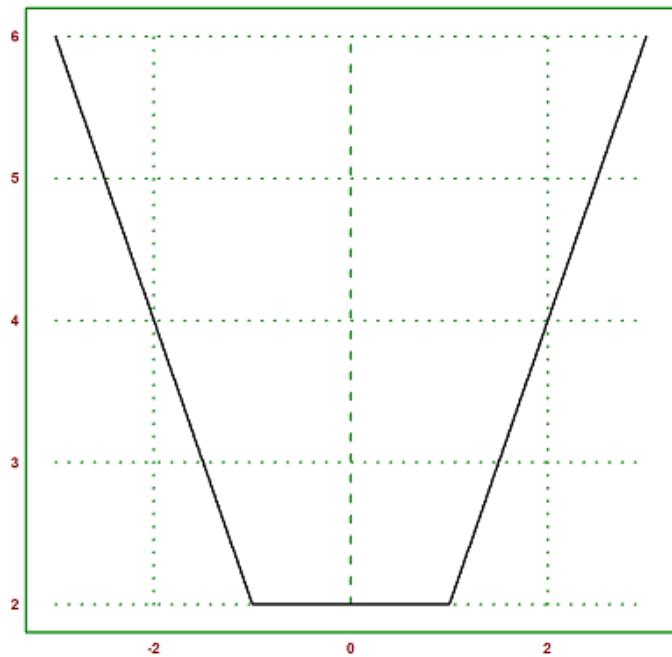
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

```
Illegal function result in map.
%ploteval2:      return map(f$,x,y,args());
f3dplot:    z=%ploteval2(f$,x,y,maps,args());
plot3d:          wire,xlabel,ylabel,zlabel,sliced,maps,wirecolor,args())
```

Pembatasan garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



### Poin Ketiga

Sekarang hal-hal yang kurang sederhana: Ini sedikit kurang terkenal bahwa  $MA+MB+MC$  mencapai minimum pada satu titik pesawat tetapi untuk menentukan itu kurang sederhana:

1) Jika salah satu sudut segitiga ABC lebih dari  $120^\circ$  (katakanlah di A), maka minimum dicapai pada titik ini (misalnya  $AB+AC$ ).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
```

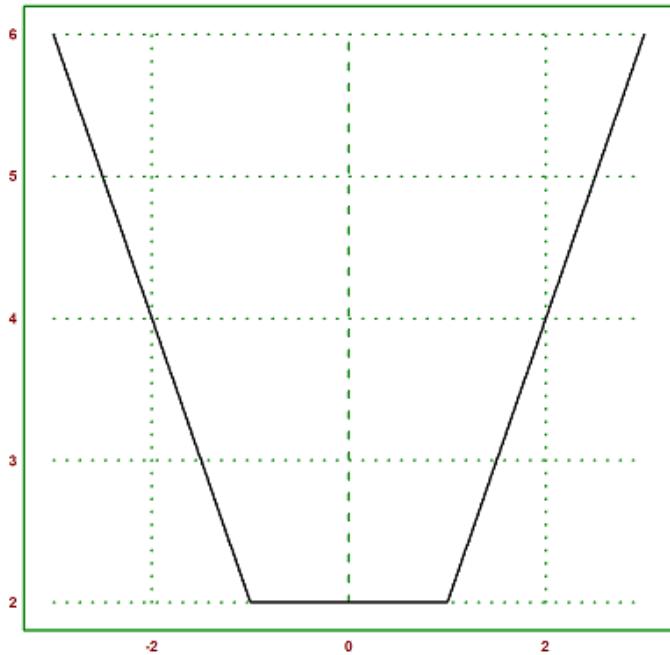
Illegal function result in map.

%ploteval2: return map(f\$,x,y,args());

f3dplot: z=%ploteval2(f\$,x,y,maps,args());

plot3d: wire,xlabel,ylabel,zlabel,sliced,maps,wirecolor,args())

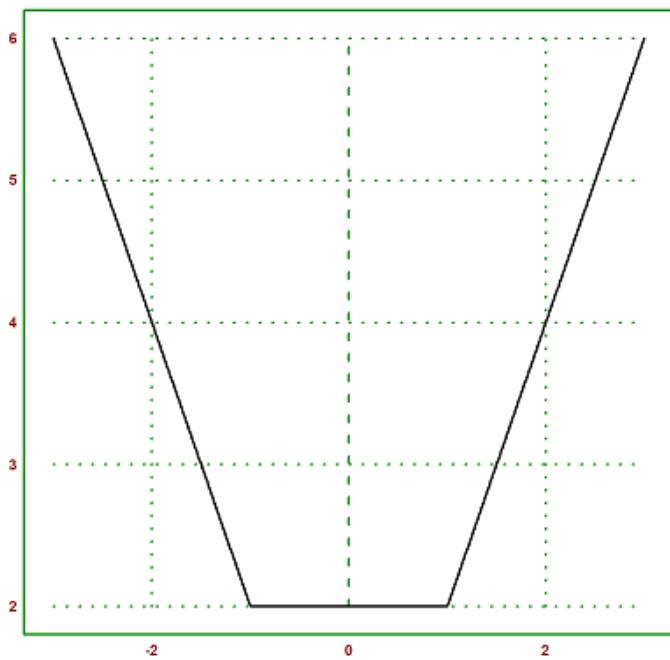
```
>insimg;
```



```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
```

```
Illegal function result in map.
%ploteval2:           return map(f$,x,y,args());
fcontour:   Z=%ploteval2(f$,X,Y,maps,args());
```

```
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



2) Tetapi jika semua sudut segitiga ABC kurang dari  $120^\circ$ , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing  $120^\circ$ ):

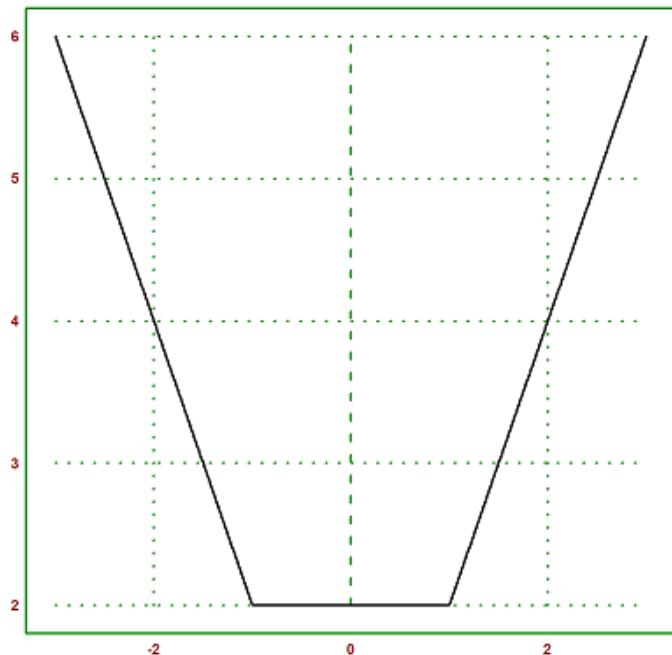
```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):

Illegal function result in map.
%ploteval2:           return map(f$,x,y,args());
f3dplot:   z=%ploteval2(f$,x,y,maps,args());
plot3d:           wire,xlabel,ylabel,zlabel,sliced,maps,wirecolor,args()

>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");

Illegal function result in map.
%ploteval2:           return map(f$,x,y,args());
fcontour:  Z=%ploteval2(f$,X,Y,maps,args());

>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimsg;
```



Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft yang ditulis di Jawa yang memiliki instruksi "garis kontur" ...

Semua ini di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada dilettants lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga  $FA+FB+FC$  minimal, disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torricelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun tradisinya adalah mencatat poin ini F...

**Poin Empat**

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```

Illegal function result in map.

%ploteval2: return map(f\$,x,y,args());

f3dplot: z=%ploteval2(f\$,x,y,maps,args());

plot3d: wire,xlabel,ylabel,zlabel,sliced,maps,wirecolor;args())

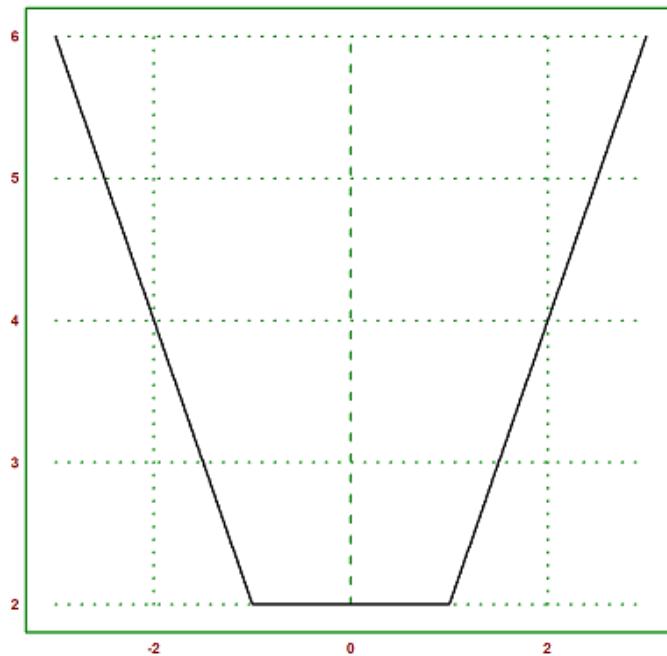
```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
```

Illegal function result in map.

%ploteval2: return map(f\$,x,y,args());

fcontour: Z=%ploteval2(f\$,X,Y,maps,args());

```
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```



Masih ada minimum dan tidak tercapai di salah satu simpul A, B, C atau D:

```
>function f(x):=d4(x[1],x[2])
>neldermin("f", [0.2,0.2])
```

Function call to "f" failed in nelder.

neldermin: v1=\_nelder(f\$,v,d,eps;args());

Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional... Sekarang ABCD adalah persegi, kami berharap bahwa titik optimal akan menjadi pusat ABCD:

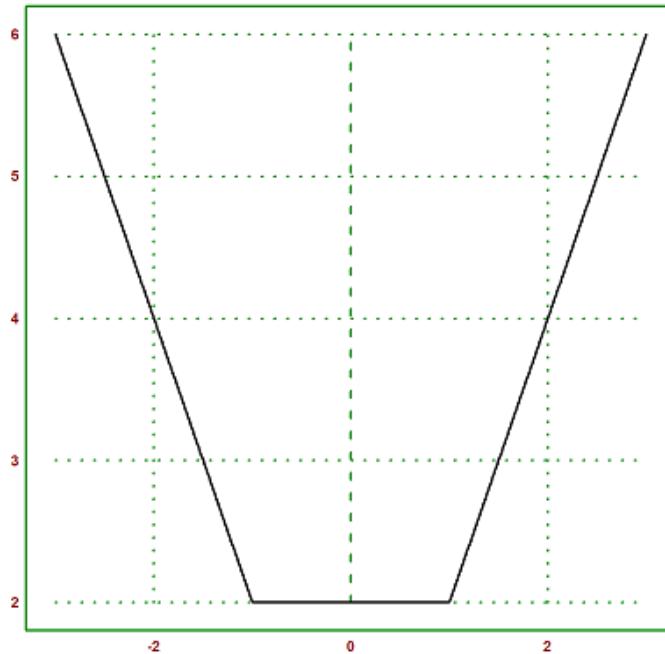
```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```

```
Illegal function result in map.
%ploteval2:           return map(f$,x,y,args());
f3dplot:      z=%ploteval2(f$,x,y,maps,args());
plot3d:                  wire,xlabel,ylabel,zlabel,sliced,maps,wirecolor,args()
```

```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
```

```
Illegal function result in map.
%ploteval2:           return map(f$,x,y,args());
fcontour:      Z=%ploteval2(f$,X,Y,maps,args());
```

```
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```



### Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan povengine.exe di jalur program.

Pertama kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometri.e dari Euler untuk ini.

```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[ - a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

```
[ - a, - 1, 0]
```

Kemudian baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

```
[ - 1, 2, 1]
```

Kami merencanakan semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);  
>color(black); plotLine(g(), "")
```

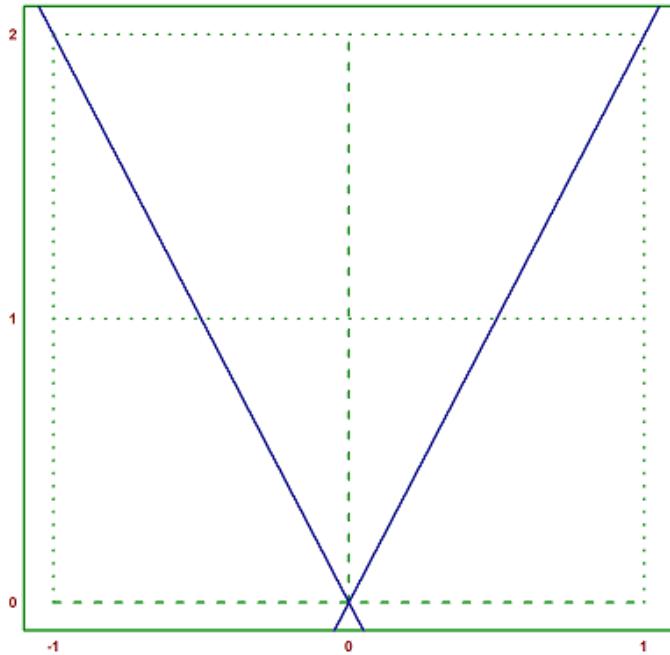
Function g needs at least one argument!

Use: g (x)

Error in :

```
color(black); plotLine(g(), "")  
^
```

```
>a:=2; color(blue); plotLine(g1(), ""), plotLine(g2(), ""):
```



Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0, u]
```

[0, u]

Hitung jarak ke g1.

```
>d1 &= distance(P, projectToLine(P, g1)); $d1
```

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

Hitung jarak ke g.

```
>d &= distance(P, projectToLine(P, g)); $d
```

$$\sqrt{\left(\frac{u+2}{5} - u\right)^2 + \frac{(2u-1)^2}{25}}$$

Dan temukan pusat kedua lingkaran yang jaraknya sama.

```
>sol &= solve(d1^2=d^2, u); $sol
```

$$\left[ u = \frac{-\sqrt{5}\sqrt{a^2 + 1} + 2a^2 + 2}{4a^2 - 1}, u = \frac{\sqrt{5}\sqrt{a^2 + 1} + 2a^2 + 2}{4a^2 - 1} \right]$$

Ada dua solusi.

Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()
```

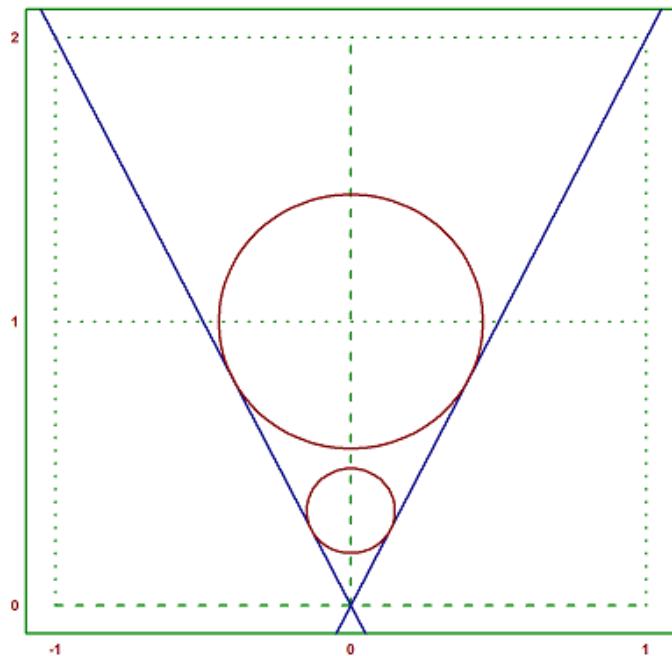
```
[0.333333, 1]
```

```
>dd := d()
```

```
[0.149071, 0.447214]
```

Plot lingkaran ke dalam gambar.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]), "");
>plotCircle(circleWithCenter([0,u[2]],dd[2]), "");
>insimsg;
```



## Plot dengan Povray

Selanjutnya kami merencanakan semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return. Pertama kita memuat fungsi povray.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Kami mengatur adegan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya kita menulis dua bidang ke file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan bidang terbatas pada kerucut.

```
>gp=g();
```

Function g needs at least one argument!  
Use: g (x)

Error in :  
gp=g();  
^

```
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0, gp[2]]; dp=gp[3];
```

gp is not a variable you can index with []!

Error in :  
vp=[gp[1],0, gp[2]]; dp=gp[3];
^

```
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Variable vp not found!

Error in :  
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
^

Sekarang kita menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]));
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]));
>writeln(povpoint(P2,povlook(yellow))));
```

Kemudian kami menghasilkan dua titik di mana bola menyentuh bidang. Ini adalah fokus dari elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
```

Function g needs at least one argument!

Use: g (x)

Error in :

```
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
^
```

```
>writeln(povpoint(P3,povlook(yellow)));
```

Variable P3 not found!

Error in :

```
writeln(povpoint(P3,povlook(yellow)));
^
```

```
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
```

Function g needs at least one argument!

Use: g (x)

Error in :

```
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
^
```

```
>writeln(povpoint(P4,povlook(yellow)));
```

Variable P4 not found!

Error in :

```
writeln(povpoint(P4,povlook(yellow)));
^
```

Selanjutnya kita hitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
```

Variable vp not found!

Error in :

```
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
^
```

```
>writeln(povpoint(P5,povlook(yellow)));
```

```
Variable P5 not found!
```

```
Error in :
```

```
writeln(povpoint(P5,povlook(yellow)));  
^
```

Kami menghubungkan titik-titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));  
>writeln(povsegment(P5,P3,povlook(yellow)));
```

```
Variable P5 not found!
```

```
Error in :
```

```
writeln(povsegment(P5,P3,povlook(yellow)));  
^
```

```
>writeln(povsegment(P5,P4,povlook(yellow)));
```

```
Variable P5 not found!
```

```
Error in :
```

```
writeln(povsegment(P5,P4,povlook(yellow)));  
^
```

Sekarang kita menghasilkan pita abu-abu, di mana bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);  
>pc1=povcylinder([0,0,P1[3]-defaultpointsiz/2],[0,0,P1[3]+defaultpointsiz/2],1);  
>writeln(povintersection([pcw,pc1],povlook(gray)));  
>pc2=povcylinder([0,0,P2[3]-defaultpointsiz/2],[0,0,P2[3]+defaultpointsiz/2],1);  
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```

```
User interrupted!  
exec: return _exec(program,param,dir,print,hidden,wait);  
povray: exec(program,params,defaulthome);  
povend: povray(file,w,h,aspect,exit);
```

Untuk mendapatkan Anaglyph ini kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali kemudian.

```
>function scene () ...
```

```

global a,u,dd,g,g1,defaultpointsiz;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,dp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsiz/2],[0,0,P1[3]+defaultpointsiz/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsiz/2],[0,0,P2[3]+defaultpointsiz/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction

```

Anda membutuhkan kacamata merah/sian untuk menghargai efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

```

exec:    return _exec(program,param,dir,print,hidden,wait);
povray:   exec(program,params,defaulthome);
povanaglyph:   povray(currentfile,w,h,aspect,exit);

```

## Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi spherical).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

S  $7^{\circ}46.467'$  E  $110^{\circ}23.050'$

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];  
>sposprint(Solo), sposprint(Semarang),
```

S  $7^{\circ}34.333'$  E  $110^{\circ}49.683'$

S  $6^{\circ}59.050'$  E  $110^{\circ}24.533'$

Pertama kita menghitung vektor dari satu ke yang lain pada bola ideal. Vektor ini [pos,jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang  $7^{\circ}$ .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth( $7^{\circ}$ )->km // perkiraan jarak FMIPA-Solo
```

$65^{\circ}20'26.60''$

53.8945384608

Ini adalah perkiraan yang baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km", // perkiraan jarak FMIPA-Semarang
```

88.0114026318 km

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

$65.34^{\circ}$

Sudut segitiga melebihi  $180^{\circ}$  pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo);
```

$180^{\circ}0'10.77''$

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>(asum-pi)*rearth( $48^{\circ}$ )^2->" km^2", // perkiraan luas segitiga FMIPA-Solo-Semarang,
```

2116.02948749 km $^2$

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

2123.64310526 km<sup>2</sup>

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor saddr.

```
>v=svector(FMIPA,Solo); sposprint(saddrvector(FMIPA,v)), sposprint(Solo),
```

S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];  
>sposprint(Tugu), sposprint(Monas)
```

S 7°46.998' E 110°21.966'  
S 6°10.500' E 106°48.717'

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

```
>esdist(Tugu,Monas)->" km", // perkiraan jarak Tugu Jogja - Monas Jakarta
```

431.565659488 km

Judulnya sama dengan judul yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

294°17'2.85''

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S  $6^{\circ}10.500'$  E  $106^{\circ}48.717'$

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

S  $6^{\circ}10.500'$  E  $106^{\circ}48.717'$

Tentu kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh.

```
>sposprint(p), skmpprint(esdist(p,Monas))
```

S  $6^{\circ}11.250'$  E  $106^{\circ}48.372'$   
1.529km

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang  $30^{\circ}$ , melainkan jalur terpendek yang dimulai  $10^{\circ}$  lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

$79.69^{\circ}$

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada  $1/10$  dari total jarak.

```
>p=P1; dist=esdist(P1,P2); ...
> loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°
81.67°
83.71°
85.78°
87.89°
90.00°
92.12°
94.22°
96.29°
98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

```
>skmpprint(esdist(p,P2))
```

```
0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
> loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmpprint(esdist(p,Monas))
```

```
0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
> loop 1 to rows(v); sposprint(v[#]), end;
```

```
S 7°46.998' E 110°21.966'
S 7°37.422' E 110°0.573'
S 7°36.357' E 109°58.197'
S 7°35.026' E 109°55.227'
S 7°33.313' E 109°51.409'
S 7°31.029' E 109°46.320'
S 7°27.829' E 109°39.196'
S 7°23.026' E 109°28.513'
S 7°15.011' E 109°10.717'
S 6°58.948' E 108°35.157'
S 6°10.500' E 106°48.717'
```

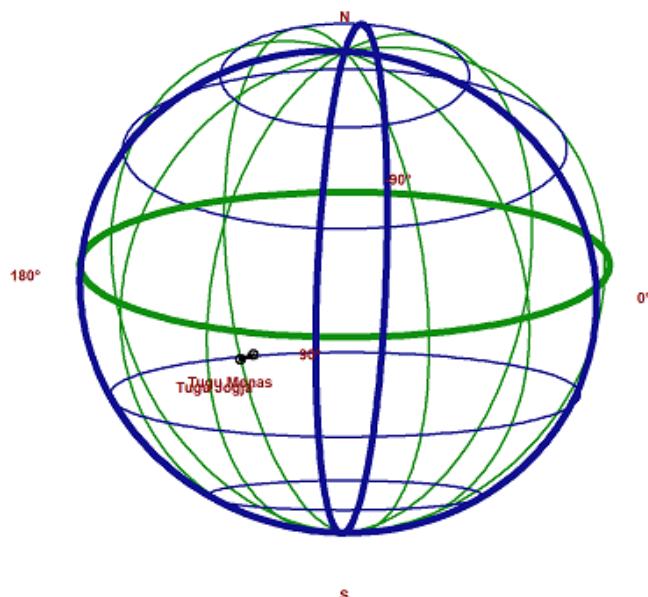
Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(Tugu, "Tugu Jogja"); plotpos(Monas, "Tugu Monas");
plotposline(v);
endfunction
```

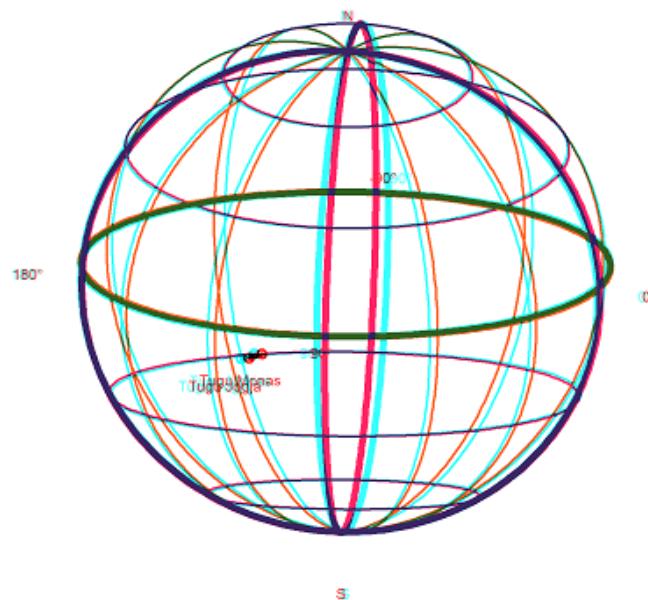
Sekarang rencanakan semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/sian.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



## Latihan

---

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

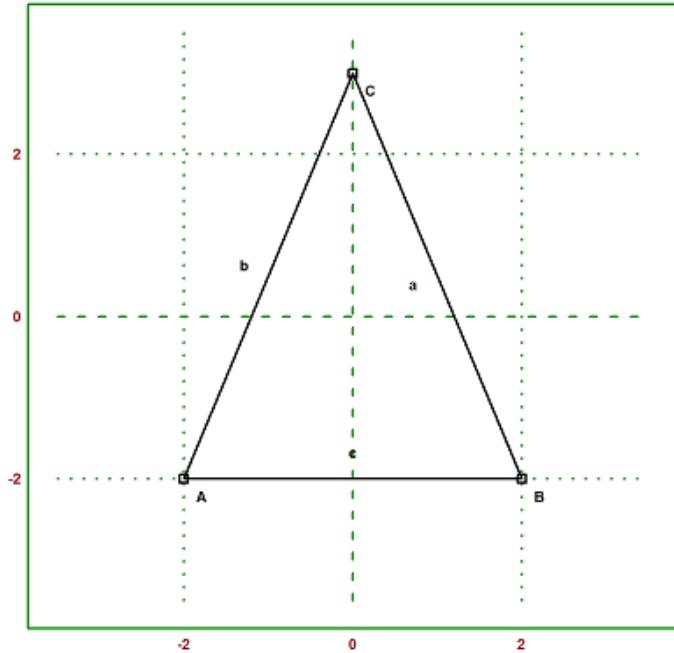
Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah  $(360/n)$ . - Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan  $(360/n)$ . - Untuk n ganjil, pilih salah satu titik sudut adalah di atas. - Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat. - Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

```
>load geometry
```

Numerical and symbolic geometry.

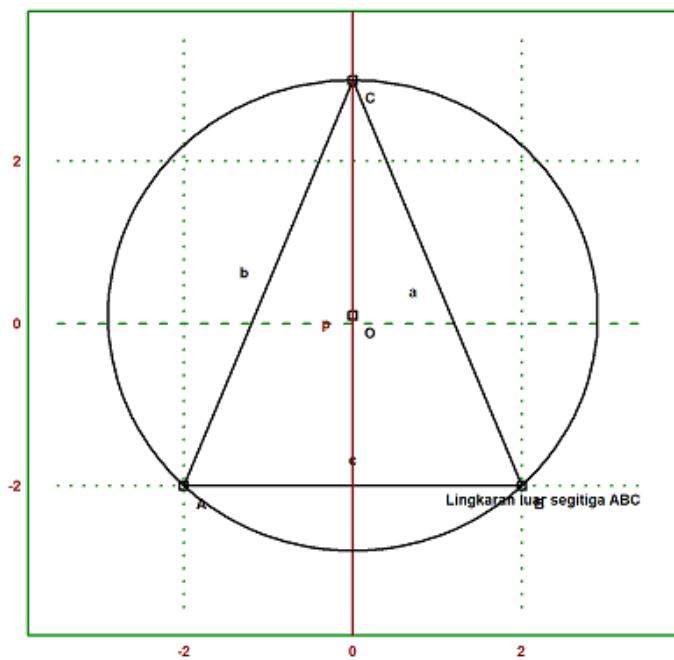
```
>setPlotRange(-3.5,3.5,-3.5,3.5);
>A=[-2,-2]; plotPoint(A,"A");
>B=[2,-2]; plotPoint(B,"B");
>C=[0,3]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
>aspect(1):
```



```

>c=circleThrough(A,B,C);
>R=getCircleRadius(c);
>O=getCircleCenter(c);
>plotPoint(O,"O");
>l=angleBisector(A,C,B);
>color(2); plotLine(l); color(1);
>plotCircle(c,"Lingkaran luar segitiga ABC"):

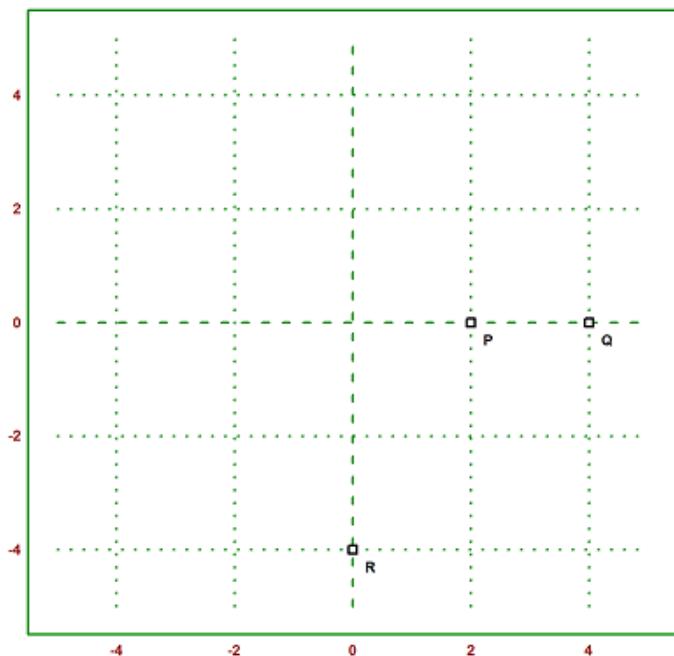
```



2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk: - Misalkan persamaan parabolanya  $y = ax^2 + bx + c$ . - Substitusikan koordinat titik-titik yang diketahui ke persamaan. Selesaikan SPL yang terbentuk untuk mendapatkan nilai  $a$ ,  $b$ ,  $c$ .

```
>load geometry;
>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");
```



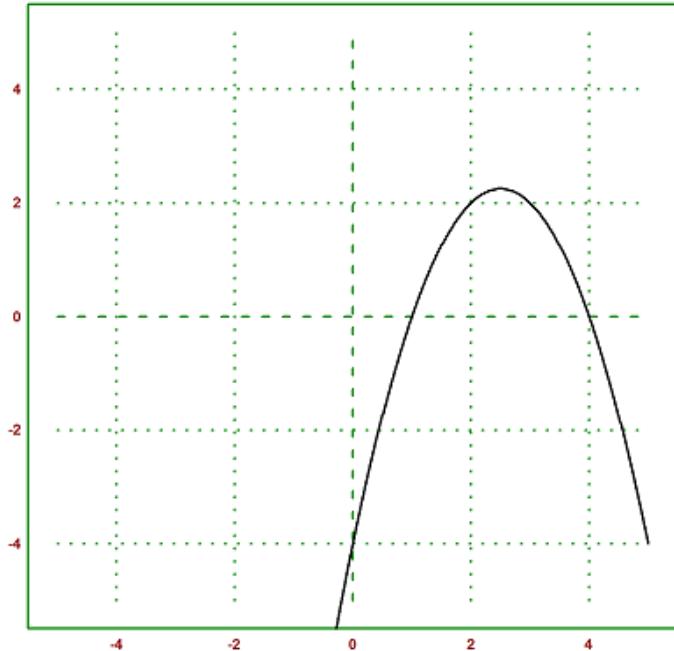
```
>sol &= solve([a+b=-c, 16*a+4*b=-c, c=-4], [a,b,c])
```

```
[[a = - 1, b = 5, c = - 4]]
```

```
>function y&=-x^2+5*x-4
```

$$-\frac{2}{x^2} + \frac{5}{x} - \frac{4}{4}$$

```
>plot2d("-x^2+5*x-4", -5, 5, -5, 5);
```

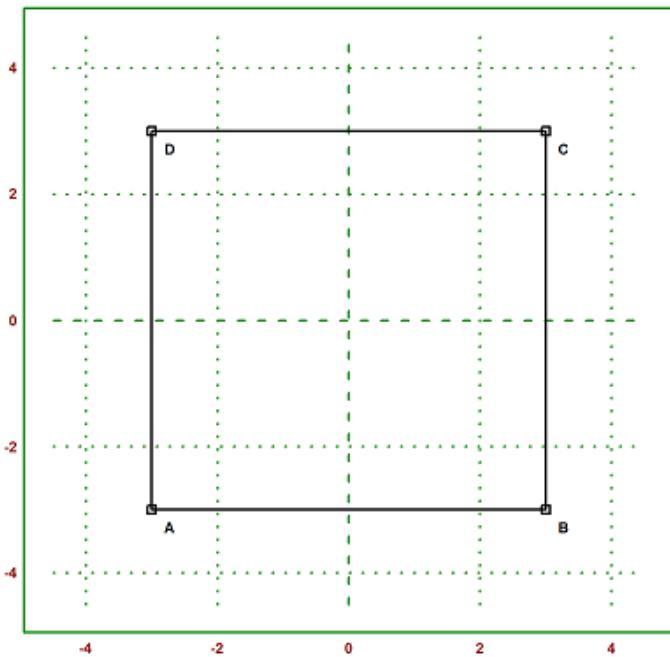


3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D. - Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisinya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut). - Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik. - Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya. - Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

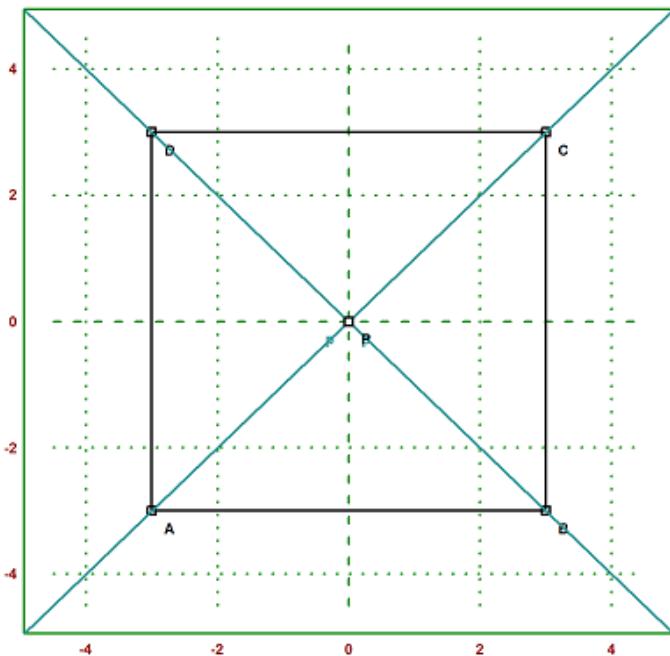
```
>load geometry
```

Numerical and symbolic geometry.

```
>setPlotRange(-4.5,4.5,-4.5,4.5);
>A=[-3,-3]; plotPoint(A,"A");
>B=[3,-3]; plotPoint(B,"B");
>C=[3,3]; plotPoint(C,"C");
>D=[-3,3]; plotPoint(D,"D");
>plotSegment(A,B,"");
>plotSegment(B,C,"");
>plotSegment(C,D,"");
>plotSegment(A,D,"");
>aspect(1):
```

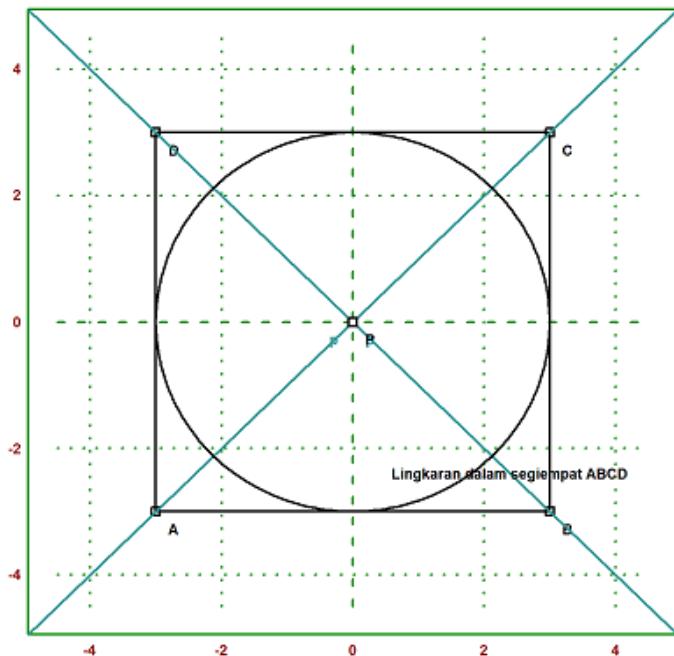


```
>l=angleBisector(A,B,C);
>m=angleBisector(B,C,D);
>P=lineIntersection(l,m);
>color(5); plotLine(l); plotLine(m); color(1);
>plotPoint(P,"P");
```



Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)));
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD");
```



Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat. Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

```
>AB=norm(A-B) //panjang sisi AB
```

6

```
>CD=norm(C-D) //panjang sisi CD
```

6

```
>AD=norm(A-D) //panjang sisi AD
```

6

```
>BC=norm(B-C) //panjang sisi BC
```

```
>AB.CD
```

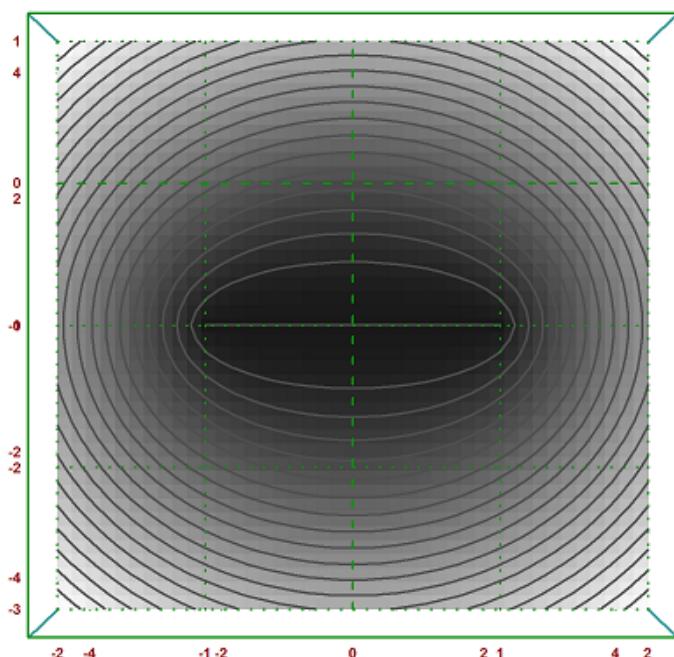
&gt;AD . BC

Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

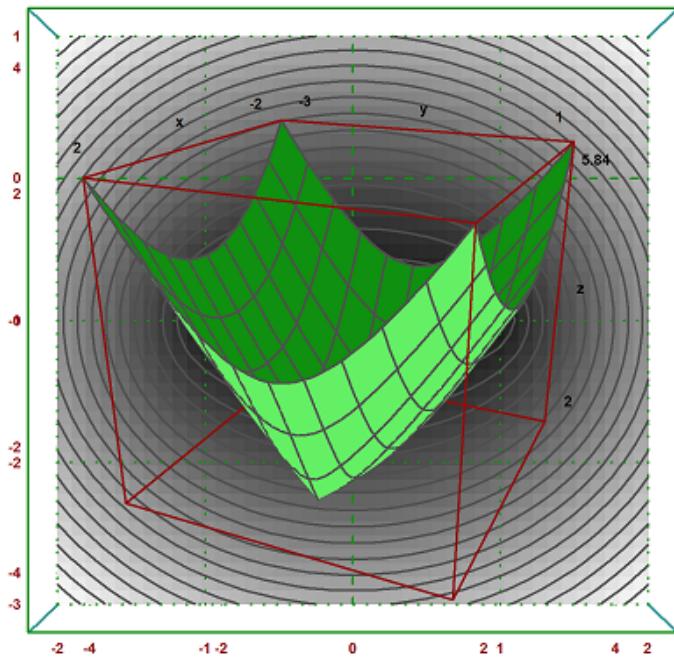
Penyelesaian : Diketahui kedua titik fokus  $P = [-1, -1]$  dan  $Q = [1, -1]$

```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x-Q[1])^2+(y-Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



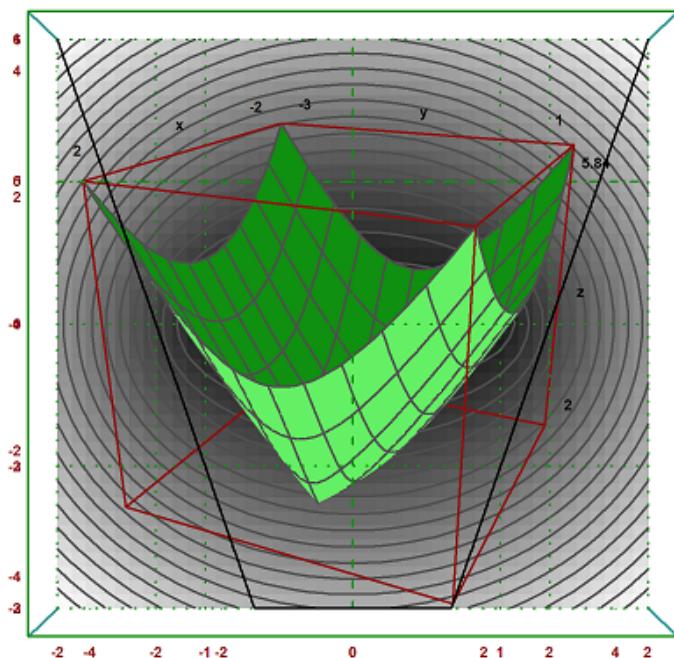
Grafik yang lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Batasan ke garis PQ

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

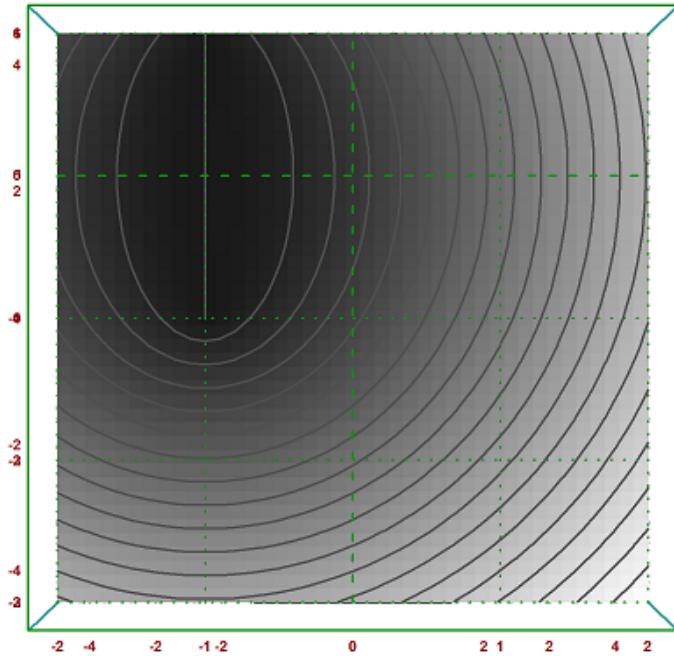


5. Gambarlah suatu hiperbolika jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

```

>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-p[1])^2+(y-p[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x+Q[1])^2+(y+Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):

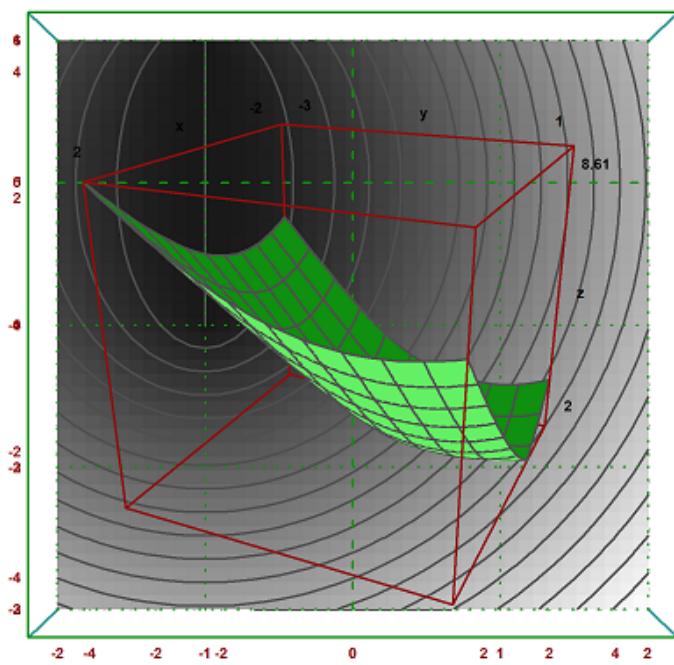
```



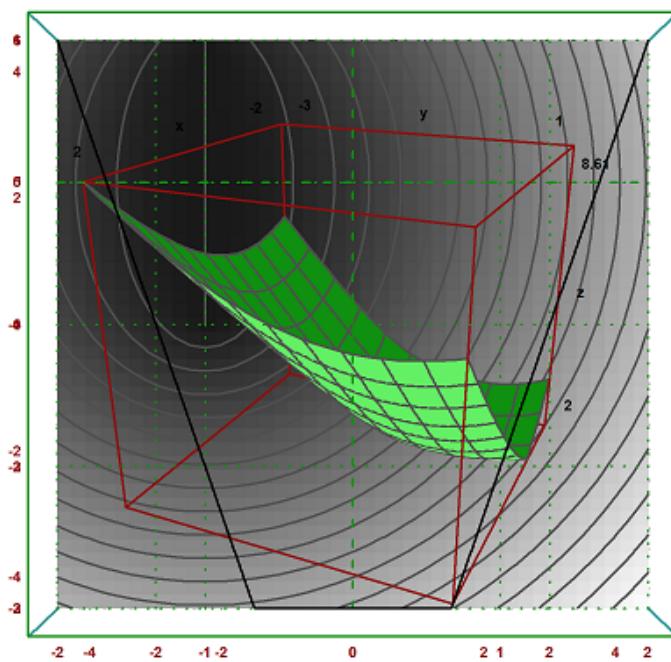
```

>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):

```



```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



---

---

## BAB 7

---

# KB PEKAN 13-14: MENGGUNAKAN EMT UNTUK STATISTIKA

### **EMT Untuk Visualisasi Dan Komputasi Statistika**

---

#### **Sub Topik 1: Menyimpan Data Dalam Bentuk Matriks**

---

##### **Array**

Array adalah kumpulan-kumpulan variabel yang menyimpan data dengan tipe yang sama atau data-data yang tersusun secara linear dimana di dalamnya terdapat elemen dengan tipe yang sama.

Vektor digunakan untuk menggambarkan array angka satu dimensi. Vektor memiliki panjang, yang merupakan jumlah elemen dalam array.

Sedangkan matriks digunakan dalam mendeskripsikan susunan bilangan dua dimensi yang disusun dalam baris dan kolom. matriks memiliki ukuran, yaitu jumlah baris dan kolom.

Hubungan antara array dan matriks adalah bahwa matriks adalah bentuk khusus dari array. Array dapat memiliki lebih dari dua dimensi, tetapi matriks selalu memiliki dua dimensi. Dalam pemrograman, array dan matriks sering digunakan untuk menyimpan data dalam jumlah besar dan memudahkan pengaksesan data tersebut.

Mari kita bahas beberapa hal terkait vektor terlebih dahulu

```
>v=shuffle(1:10)
```

```
[6, 3, 1, 5, 10, 4, 9, 8, 2, 7]
```

```
>w=intrandom(10,12)
```

```
[11, 4, 9, 3, 6, 4, 11, 3, 6, 2]
```

Untuk mengurutkan angka acak

```
>sort(v)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Selanjutnya mengurutkan angka acak dengan menyederhanakan angka yang sama

```
>unique(v)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Menemukan banyaknya setiap elemen dengan bantuan interval

```
>s=intrandom(10,20)
```

```
[12, 9, 15, 10, 7, 11, 11, 4, 6, 18]
```

```
>x=[5,10,15,20]
```

```
[5, 10, 15, 20]
```

```
>find(x,s)
```

```
[2, 1, 3, 2, 1, 2, 2, 0, 1, 3]
```

Berikutnya adalah cara mencari indeks dari sebuah vektor dengan contoh vektor indeks pada EMT berbeda dengan indeks pada Phyton yang kita pelajari sebelumnya di Algoritma dan pemrograman. Perbedaannya jika sebelumnya untuk menentukan indeks akan dimulai dari nol namun di EMT menentukan indeks di EMT akan dimulai dari angka satu, berikut penjelasannya

```
>indexof(w,1:10)
```

```
[0, 10, 4, 2, 0, 5, 0, 0, 3, 0]
```

```
>x= sort(intrandom(10,12))
```

```
[1, 1, 3, 6, 7, 8, 8, 12, 12, 12]
```

```
>indexofsorated(x,1:15)
```

```
[2, 0, 3, 0, 0, 4, 5, 7, 0, 0, 0, 10, 0, 0, 0]
```

```
>z=intrandom(1000,10); multofsoted(sort(z),1:10), sum(%)
```

```
[81, 96, 121, 101, 102, 91, 115, 98, 100, 95]  
1000
```

Sampai disini pembahasan terkait dengan vektor

Selanjutnya kita akan membahas beberapa hal terkait matriks terkait

Untuk Menyimpan Data dalam bentuk Matrik

Pertama, buat sebuah variabel yang akan menampung data matrik, misal X. Variabel ini bebas dengan syarat tidak sama dengan nama fungsi atau konstanta yang sudah ada dalam software.

Selanjutnya,kita akan membuat matrik berordo mxn yang berisi angka

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

1	2	3	4
4	5	6	7
8	4	4	6

```
>shortformat; A=random(3,4)
```

0.50136	0.58172	0.02845	0.72032
0.27668	0.1313	0.84982	0.77608
0.32956	0.68574	0.42373	0.77217

```
>shortformat; A=intrandom(5,4,20)
```

3	18	4	18
7	19	12	8
5	17	11	10
15	1	20	7
11	13	9	2

```
>shortformat; A=redim(1:15,4,4)
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	0

```
>(1:5)_2
```

1	2	3	4	5
2	2	2	2	2

```
>random(3,3)_random(2,2)
```

```

0.74041  0.41322  0.25054
0.15543  0.10655  0.98859
0.48475  0.078167 0.57911
0.46856  0.22056   0
0.22837  0.47473   0

```

```
>for k=1 to prod(size(A)); A{k}=k; end; short A
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

```
>B=zeros(size(A))
```

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

```
>B=ones(size(A))
```

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

Berikutnya operasi penjumlahan dan pengurangan matriks

```
>shortformat; I=intrandom(3,4,10)
```

5	5	7	5
9	8	3	8
5	5	1	1

```
>shortformat; J=intrandom(3,4,8)
```

7	8	1	3
3	7	6	7
1	1	1	6

```
>C= I-J
```

-2	-3	6	2
6	1	-3	1
4	4	0	-5

```
>C= I+J
```

12	13	8	8
12	15	9	15
6	6	2	7

Dalam materi matriks yang pernah kita pelajari ada sebutan transpose, Invers dan juga determinan, jika menggunakan EMt sebagai berikut secara berurutan:

```
>T = transpose(I)
```

5	9	5
5	8	5
7	3	1
5	8	1

```
>T = I'
```

5	9	5
5	8	5
7	3	1
5	8	1

```
>K = J^(-1)
```

0.14286	0.125	1	0.33333
0.33333	0.14286	0.16667	0.14286
1	1	1	0.16667

```
>shortformat; L=intrandom(3,3,7)
```

7	4	4
3	1	6
3	7	6

```
>det(L)
```

-180

Selanjutnya adalah cara ekstraksi baris dan kolom, atau sub-matriks,yang mirip dengan R sebagai berikut:

```
>L[,2:3]
```

4	4
1	6
7	6

```
>shortformat; X=redim(1:20,4,5)
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

```
>function setmatrixvalue (M, i, j, v) ...
```

```
loop 1 to max(length(i),length(j),length(v))
  M[i{#},j{#}] = v{#};
end;
endfunction
```

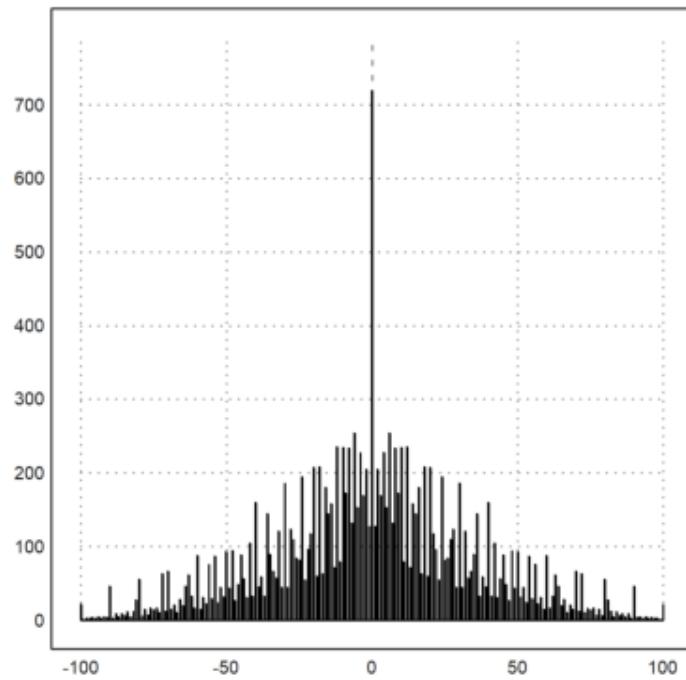
```
>setmatrixvalue(X,1:4,4:-1:1,0); X,
```

1	2	3	0	5
6	7	0	9	10
11	0	13	14	15
0	17	18	19	20

```
>(1:4)*(1:4)'
```

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

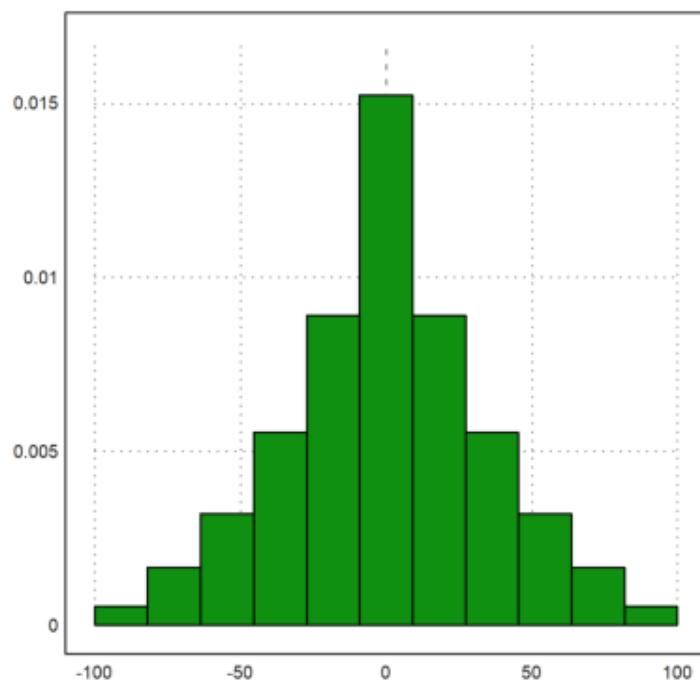
```
>a=0:10; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
>u=sort(unique(q)); f=getmultiplicities(u,q); ...
>statplot(u,f,"h"):
```



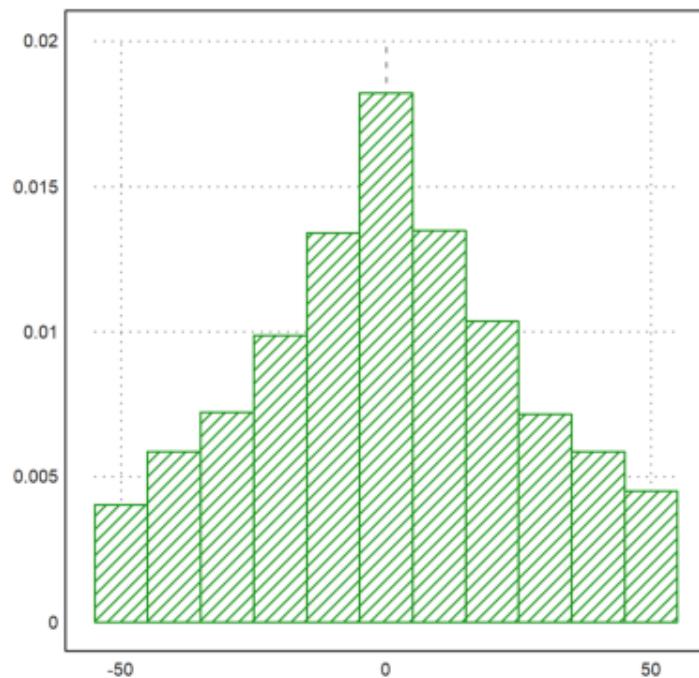
```
>getfrequencies(q,-50:10:50)
```

```
[613, 814, 1088, 1404, 1904, 2389, 1431, 1109, 841, 680]
```

```
>plot2d(q,distribution=11):
```



```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x);
>plot2d(x,y,>bar,style="/"):
```



## Sub Topik 2 : Menghasilkan Data Acak Menggunakan Fungsi Distribusi

CAKUPAN MATERI MELIPUTI DIANTARANYA:

- Definisi Bilangan Acak dan Data Acak
  - Pengertian Distribusi Diskrit dan Konsep yang Terkait
  - Metode Menentukan Distribusi Diskrit
1. Definisi Bilangan Acak dan Data Acak

Bilangan Acak adalah bilangan yang tidak dapat diprediksi

kemunculannya. Sehingga, tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna.

Bilangan acak sendiri dapat dibangkitkan dengan pola tertentu yang dinamakan dengan distribusi, dengan catatan mengikuti fungsi distribusi yang ditentukan.

Data acak merupakan hasil dari suatu percobaan acak. Sedangkan percobaan acak adalah suatu proses yang dilakukan sedemikian rupa sehingga hasilnya tidak dapat ditentukan dengan pasti sebelum percobaan tersebut selesai dilakukan  
contoh :

```
>intrandom(1,10,10)
```

```
[2, 4, 6, 7, 3, 3, 2, 9, 10, 2]
```

---

## 2. Pengertian Distribusi Diskrit dan Konsep yang Terkait

Distribusi diskrit dalam statistika adalah distribusi data yang memiliki nilai-nilai yang terpisah dan dapat dihitung. Contohnya adalah jumlah anak dalam sebuah keluarga, jumlah mata dadu yang muncul, atau jumlah pelanggan yang datang ke sebuah toko.

Distribusi diskrit merujuk pada distribusi probabilitas yang melibatkan variabel acak diskrit. Variabel acak diskrit adalah variabel acak yang hanya dapat mengambil nilai-nilai terpisah, bukan nilai-nilai kontinu seperti pada variabel acak kontinu. Distribusi diskrit memberikan probabilitas masing-masing nilai yang mungkin dari variabel acak tersebut. Berikut adalah beberapa konsep kunci yang terkait dengan distribusi diskrit dalam statistika:

1. Fungsi Probabilitas Diskrit (Probability Mass Function - PMF):  
-Fungsi probabilitas diskrit, atau PMF, memberikan probabilitas bahwa variabel acak diskrit akan mengambil nilai tertentu.  
PMF umumnya dilambangkan dengan  $P(X=x)$ , di mana X adalah variabel acak dan x adalah nilai yang mungkin dari variabel tersebut.

### 2. Ruang Sampel (Sample Space):

- Ruang sampel adalah himpunan semua hasil mungkin dari suatu percobaan acak yang dapat diukur.
- Setiap elemen dalam ruang sampel merupakan hasil yang mungkin dari variabel acak.

### 3. Hukum Probabilitas untuk Distribusi Diskrit:

Probabilitas suatu kejadian adalah bilangan yang berada dalam rentang 0 hingga 1, atau  $0 \leq P(A) \leq 1$  untuk setiap kejadian A.

Probabilitas total dari semua hasil dalam ruang sampel adalah 1, atau  $P(S) = 1$ , di mana S adalah ruang sampel.

#### 4. Fungsi Distribusi Kumulatif (Cumulative Distribution Function - CDF):

- Fungsi distribusi kumulatif memberikan probabilitas bahwa variabel acak diskrit kurang dari atau sama dengan nilai tertentu.
- Notasi matematisnya sering kali disimbolkan sebagai  $F(x) = P(X \leq x)$

#### 5. Harapan (Expectation) dan Varians:

- Harapan atau nilai rata-rata ( $E(X)$ ) dari distribusi diskrit adalah jumlah tertimbang dari nilai-nilai mungkin berdasarkan probabilitas masing-masing nilai.
- Varians  $\text{Var}(X)$  mengukur sejauh mana nilai-nilai distribusi tersebar dari nilai rata-ratanya.

>

---

### 3. Metode Menentukan Distribusi Diskrit

Untuk menentukan distribusi diskrit sendiri, dapat menggunakan metode berikut. Pertama kita mengatur fungsi distribusi, fungsi distribusi adalah fungsi yang menggambarkan kemungkinan suatu variabel acak untuk memiliki nilai tertentu atau dalam rentang waktu tertentu. Langkah mengatur fungsi distribusi:

- Menentukan jenis variabel acak yang akan diteliti, apakah diskrit atau kontinu
- Menentukan parameter-parameter yang berkaitan dengan fungsi distribusi, spt probabilitas
- Menentukan bentuk fungsi distribusi yg sesuai dg variabel acak dan parameter yg sudah ditentukan

```
>wd = 0 | ((1:6)+[-0.01,0.01,0,0,0,0])/5
```

[0, 0.198, 0.402, 0.6, 0.8, 1, 1.2]

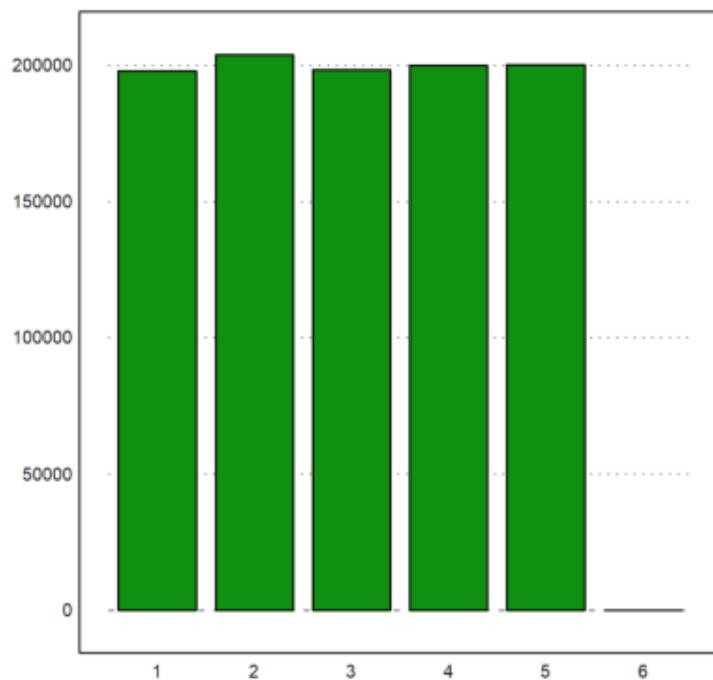
Artinya dengan probabilitas  $wd[i+1]-wd[i]$  kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator angka acak untuk ini. Fungsi `find(v,x)` menemukan nilai x dalam vektor v. Fungsi ini juga berlaku untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya sangat halus sehingga melihatnya hanya dengan iterasi yang sangat banyak.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai 1...K dalam v. menerima hasilnya, jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1)
```

```
K=max(v); n=cols(v);
fr=getfrequencies(v,1:K);
return max(fr/n-1/K)<delta/sqrt(n);
endfunction
```

Memang fungsi menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan itu menerima generator acak bawaan.

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kita dapat menghitung distribusi binomial. Pertama ada `binomials()`, yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

```
>bindis(410,1000,0.4)
```

0.7514

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Level default adalah alfa.

Arti interval ini adalah jika p berada di luar interval, hasil pengamatan 410 dalam 1000 jarang terjadi.

```
>clopperpearson(410,1000)
```

[0.37932, 0.44121]

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Tapi untuk n besar, penjumlahan langsungnya tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

0.7514

`invbins()` menghitung kebalikan dari `binomials()`.

```
>2*hypergeomsum(1,5,13,26)
```

0.32174

Ada juga simulasi distribusi multinomial. Distribusi diskrit dalam statistika adalah distribusi data yang memiliki nilai-nilai yang terpisah dan dapat dihitung. Contohnya adalah jumlah anak dalam sebuah keluarga, jumlah mata dadu yang muncul, atau jumlah pelanggan yang datang ke sebuah toko.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

385	95	520
407	88	505
388	106	506
381	114	505
413	113	474
388	99	513
396	91	513
399	93	508
390	89	521
402	103	495

### Contoh Soal

Simulasikan 1000 data acak dengan distribusi normal dengan mean 1 dan simpangan baku 2. Hitung rata-rata!

Jawab :

```
// Simulasi data acak dengan distribusi normal
```

```
>data = rrandnormal(1,1000,2)
```

```
[3.9181, 1.9527, -0.33732, 2.2526, 1.221, 1.5064, 1.0475,  
2.1997, 2.5056, 1.9128, 1.727, 2.1483, 2.2224, 3.6901, 1.1749,  
1.9995, 0.24513, 2.3104, 3.1337, 0.38286, 2.8783, 1.2136,  
2.0146, -0.097474, 2.1818, 0.0020967, 3.6067, 2.2221, 1.6357,  
0.73029, 3.1022, 2.1763, 2.6671, 2.3198, 2.326, 1.5059, 1.4371,  
2.7432, 0.1283, 1.6684, 1.9932, 0.76937, 2.134, 1.7466, 1.3792,  
1.5807, 0.83863, 2.5515, 3.2569, 4.4061, 1.9229, 3.8785,  
1.5262, 2.3406, 1.8594, 2.4003, 2.8752, 2.5498, 2.2527, 2.7602,  
2.6761, 1.6431, 3.4518, 2.2219, 1.7896, 2.519, 2.2191, 2.1538,  
2.4901, 1.8535, 3.1297, 1.7501, 4.6754, 3.3252, 1.7295, 1.4201,  
2.515, 2.5155, 0.51684, 1.727, 1.4341, 3.1193, 1.0781, 1.1891,  
0.44626, 3.7094, 1.1403, 0.064038, 0.497, 2.4713, 2.0817,  
2.2472, 4.3022, 3.5434, 2.9278, 2.5287, 1.1453, 1.9166, 2.1864,  
1.8073, 2.3455, 0.32874, 2.8625, 1.8259, 1.6132, 1.9987,  
2.8166, 4.1364, 1.2717, 3.2232, 1.1259, 1.2758, 1.5482, 3.7335,  
1.509, 3.0431, 0.60999, 1.0187, 1.8762, 1.9697, 1.6101, 1.8664,  
2.1309, 2.2315, 2.0468, 1.559, 3.853, 1.89, 1.5104, 2.1613,  
1.5506, 5.3184, 2.3794, 2.3148, 2.5744, 1.7659, 2.6473,  
0.66693, 2.6216, 4.014, 3.6484, -0.083378, 3.5684, 2.7952,  
2.1936, 2.381, 1.8078, 0.78504, 3.4552, 1.2495, 0.88303,  
... ]
```

```
>mean(data)
```

```
1.9987
```

## Sub Topik 3: Membaca Data Yang Tersimpan

Membaca Data yang Tersimpan di dalam Berkas dengan Berbagai Format(teks biasa, CSV) untuk di analisis lebih lanjut

membaca data dari teks biasa pertama-tama, kita akan mencoba membaca data dengan teks biasa yang terdapat pada contoh yang telah di berikan pada besmart yaitu dengan menuliskan fungsi "printfile(nama file teks biasa, berapa baris yang akan di print)"

seperti di bawah kita akan print tabel pada data yang terdapat pada buku online "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat", 4)
```

```
Person Sex Age Titanic Evaluation Tip Problem  
1 m 30 n . 1.80 n  
2 f 23 y g 1.80 n  
3 f 26 y g 1.80 y
```

pada kasus ini saya telah menunjukkan terdapat 4 baris yang telah di print pada emt yang berisi angka dan token(string)

kali ini kita akan membaca tabel dengan lebih mudah atau dengan bahasa kita sendiri. Untuk ini, kita mendefinisikan set token. Fungsi strtokens () mendapatkan vektor string token dari string tertentu.

```
>mf:=[ "m", "f"]; yn:=[ "y", "n"]; ev:=strtokens("g vg m b vb");
```

sekarang kita dapat membacanya dengan cara kita sendiri

Argumen tok2, tok4, dll. Adalah definisi dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable (), jadi Anda perlu memberinya ":" untuk mendefinisikannya.

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);  
>load over statistics;
```

lalu kita akan print tabel sesuai dengan tabel awal namun dengan bentuk tabel yang berbeda

```
>writetable(MT[1:6],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y

Titik "." mewakili nilai-nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan terlebih dahulu, kita hanya perlu menentukan, kolom mana yang berisi token dan bukan angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

Fungsi readtable () sekarang mengembalikan satu set token.

```
>tok
```

```
m  
n  
f  
y  
g  
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan menjadi angka.

String khusus NA = "." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
>MT [1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut adalah isi tabel dengan bilangan yang belum diterjemahkan.

```
>writetable(MT[1:6], wc=5)
```

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4

Untuk kenyamanan, Anda bisa memasukkan keluaran readtable () ke dalam daftar.

```
>Table={{readtable("table.dat", ctok=ctok)} };
```

Dengan menggunakan kolom token yang sama dan token dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. Atau menggunakan Tabel daftar.

```
>writetable(Table, ctok=ctok, wc=5);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n
18	m	38	y	g	.	n
19	f	29	n	.	3.8	n
20	f	28	y	vg	1.8	n

```

21   f   28      y      m   2.8      y
22   f   28      y      vg  1.8      y
23   f   38      y      g   2.8      n
24   f   27      y      m   1.8      y
25   m   27      n      .   2.8      y

```

tabel sudah dapat di analisis lebih lanjut

## membaca data dari CSV

---

pertama tama kita download file csv yang telah di sediakan di besmart, setelah itu kita jadi satukan dalam 1 folder dengan file emt kita. lalu masukan file tersebut dengan definisi file="nama file csv"

```

>file="test.csv"; ...
>M=random(3,3); writematrix(M,file)

```

M mendefinisikan sebagai matrix

random(n,m) mendefinisikan matrix dengan variabel acak yang akan di keluarakan  
writematrix digunakan untuk menuliskan matriks yang ada

lalu kita print datanya dengan

```
>printfile(file)
```

```

0.4191983703672241,0.4504034185243261,0.8530870403686531
0.7175631662797505,0.6559204322999515,0.883095901954524
0.07277738544441656,0.8168773103666358,0.5703787256414963

```

titik desimal pada data tersebut dapat di jadikan pada format EMT dengan cara menggunakan readmatrix()

```
>readmatrix(file)
```

```

0.4192    0.4504    0.85309
0.71756   0.65592   0.8831
0.072777  0.81688   0.57038

```

```
>
```

Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai dipisahkan koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut ini contohnya.

```
>printfile("excel-data.csv")
```

```

0;1000;1000
1;1051,271096;1072,508181
2;1105,170918;1150,273799
3;1161,834243;1233,67806
4;1221,402758;1323,129812

```

```
5;1284,025417;1419,067549  
6;1349,858808;1521,961556  
7;1419,067549;1632,31622  
8;1491,824698;1750,6725  
9;1568,312185;1877,610579  
10;1648,721271;2013,752707
```

Seperti yang Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak perlu membaca matriks ke EMT. Cara termudah untuk membaca ini ke dalam Euler adalah readmatrix (). Semua koma diganti dengan titik dengan parameter > koma. Untuk CSV bahasa Inggris, cukup abaikan parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

0	1000	1000
1	1051.3	1072.5
2	1105.2	1150.3
3	1161.8	1233.7
4	1221.4	1323.1
5	1284	1419.1
6	1349.9	1522
7	1419.1	1632.3
8	1491.8	1750.7
9	1568.3	1877.6
10	1648.7	2013.8

data siap di analisis lebih lanjut

```
>reset;
```

## Latihan

### nomer 1

```
>file="sample.csv"
```

sample.csv

```
>printfile(file,7)
```

```
female,read,write,math,hon,femalexmath  
0,57,52,41,0,0  
1,68,59,53,0,53  
0,44,33,54,0,0  
0,63,44,47,0,0  
0,47,52,57,0,0  
0,44,52,51,0,0
```

```
>ctok=[1]; {MT,hd,tok}=readtable("sample.csv",ctok=ctok);  
>tok
```

```
0  
1
```

```
>MT [2]
```

```
[2, 68, 59, 53, 0, 53]
```

```
>writetable(MT [1:6],wc=5)
```

1	57	52	41	0	0
2	68	59	53	0	53
1	44	33	54	0	0
1	63	44	47	0	0
1	47	52	57	0	0
1	44	52	51	0	0

```
>Table={{readtable(file,ctok=ctok)} };  
>writetable(Table,ctok=ctok,wc=10);
```

female	read	write	math	hon	femalexmath
0	57	52	41	0	0
1	68	59	53	0	53
0	44	33	54	0	0
0	63	44	47	0	0
0	47	52	57	0	0
0	44	52	51	0	0
0	50	59	42	0	0
0	34	46	45	0	0
0	63	57	54	0	0
0	57	55	52	0	0
0	60	46	51	0	0
0	57	65	51	1	0
0	73	60	71	0	0
0	54	63	57	1	0
0	45	57	50	0	0
0	42	49	43	0	0
0	47	52	51	0	0
0	57	57	60	0	0
0	68	65	62	1	0
0	55	39	57	0	0
0	63	49	35	0	0
0	63	63	75	1	0
0	50	40	45	0	0
0	60	52	57	0	0
0	37	44	45	0	0
0	34	37	46	0	0
0	65	65	66	1	0

0	47	57	57	0	0
0	44	38	49	0	0
0	52	44	49	0	0
0	42	31	57	0	0
0	76	52	64	0	0
0	65	67	63	1	0
0	42	41	57	0	0
0	52	59	50	0	0
0	60	65	58	1	0
0	68	54	75	0	0
0	65	62	68	1	0
0	47	31	44	0	0
0	39	31	40	0	0
0	47	47	41	0	0
0	55	59	62	0	0
0	52	54	57	0	0
0	42	41	43	0	0
0	65	65	48	1	0
0	55	59	63	0	0
0	50	40	39	0	0
0	65	59	70	0	0
0	47	59	63	0	0
0	57	54	59	0	0
0	53	61	61	1	0
0	39	33	38	0	0
0	44	44	61	0	0
0	63	59	49	0	0
0	73	62	73	1	0
0	39	39	44	0	0
0	37	37	42	0	0
0	42	39	39	0	0
0	63	57	55	0	0
0	48	49	52	0	0
0	50	46	45	0	0
0	47	62	61	1	0
0	44	44	39	0	0
0	34	33	41	0	0
0	50	42	50	0	0
0	44	41	40	0	0
0	60	54	60	0	0
0	47	39	47	0	0
0	63	43	59	0	0
0	50	33	49	0	0
0	44	44	46	0	0
0	60	54	58	0	0
0	73	67	71	1	0
0	68	59	58	0	0
0	55	45	46	0	0
0	47	40	43	0	0
0	55	61	54	1	0
0	68	59	56	0	0
0	31	36	46	0	0
0	47	41	54	0	0
0	63	59	57	0	0
0	36	49	54	0	0
0	68	59	71	0	0

0	63	65	48	1	0
0	55	41	40	0	0
0	55	62	64	1	0
0	52	41	51	0	0
0	34	49	39	0	0
0	50	31	40	0	0
0	55	49	61	0	0
0	52	62	66	1	0
0	63	49	49	0	0
1	68	62	65	1	65
1	39	44	52	0	52
1	44	44	46	0	46
1	50	62	61	1	61
1	71	65	72	1	72
1	63	65	71	1	71
1	34	44	40	0	40
1	63	63	69	1	69
1	68	60	64	0	64
1	47	59	56	0	56
1	47	46	49	0	49
1	63	52	54	0	54
1	52	59	53	0	53
1	55	54	66	0	66
1	60	62	67	1	67
1	35	35	40	0	40
1	47	54	46	0	46
1	71	65	69	1	69
1	57	52	40	0	40
1	44	50	41	0	41
1	65	59	57	0	57
1	68	65	58	1	58
1	73	61	57	1	57
1	36	44	37	0	37
1	43	54	55	0	55
1	73	67	62	1	62
1	52	57	64	0	64
1	41	47	40	0	40
1	60	54	50	0	50
1	50	52	46	0	46
1	50	52	53	0	53
1	47	46	52	0	52
1	47	62	45	1	45
1	55	57	56	0	56
1	50	41	45	0	45
1	39	53	54	0	54
1	50	49	56	0	56
1	34	35	41	0	41
1	57	59	54	0	54
1	57	65	72	1	72
1	68	62	56	1	56
1	42	54	47	0	47
1	61	59	49	0	49
1	76	63	60	1	60
1	47	59	54	0	54
1	46	52	55	0	55
1	39	41	33	0	33

1	52	49	49	0	49
1	28	46	43	0	43
1	42	54	50	0	50
1	47	42	52	0	52
1	47	57	48	0	48
1	52	59	58	0	58
1	47	52	43	0	43
1	50	62	41	1	41
1	44	52	43	0	43
1	47	41	46	0	46
1	45	55	44	0	44
1	47	37	43	0	43
1	65	54	61	0	61
1	43	57	40	0	40
1	47	54	49	0	49
1	57	62	56	1	56
1	68	59	61	0	61
1	52	55	50	0	50
1	42	57	51	0	51
1	42	39	42	0	42
1	66	67	67	1	67
1	47	62	53	1	53
1	57	50	50	0	50
1	47	61	51	1	51
1	57	62	72	1	72
1	52	59	48	0	48
1	44	44	40	0	40
1	50	59	53	0	53
1	39	54	39	0	39
1	57	62	63	1	63
1	57	60	51	0	51
1	42	57	45	0	45
1	47	46	39	0	39
1	42	36	42	0	42
1	60	59	62	0	62
1	44	49	44	0	44
1	63	60	65	0	65
1	65	67	63	1	63
1	39	54	54	0	54
1	50	52	45	0	45
1	52	65	60	1	60
1	60	62	49	1	49
1	44	49	48	0	48
1	52	67	57	1	57
1	55	65	55	1	55
1	50	67	66	1	66
1	65	65	64	1	64
1	52	54	55	0	55
1	47	44	42	0	42
1	63	62	56	1	56
1	50	46	53	0	53
1	42	54	41	0	41
1	36	57	42	0	42
1	50	52	53	0	53
1	41	59	42	0	42
1	47	65	60	1	60

```
1      55      59      52      0      52
1      42      46      38      0      38
1      57      41      57      0      57
1      55      62      58      1      58
1      63      65      65      1      65
```

```
>reset;
```

nomer 2

```
>file="test.dat"
```

```
test.dat
```

```
>printfile(file,7)
```

```
A,B,C
0.5403081010683386,0.9548801091335757,0.1530175968103556
0.8596893833361555,0.975812107764198,0.0443954899784255
0.6000027218834643,0.6869600249415273,0.7555738981923933
```

```
>mf:=[ "m", "f" ];
>{MT,hd}=readtable("table1.dat",tok2:=mf);
>load over statistics;
>writetable(MT[1:6],labc=hd,wc=5,tok2:=mf);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2

```
>reset;
```

## Sub Topik 4: Membaca data dari internet

Membaca data dari internet

Situs web atau file dari URL dapat dibuka dengan menggunakan EMT dan dapat dibaca baris demi baris.

Berikut contoh penggunaan EMT untuk membuka url untuk mengetahui versi dari EMT

```
>function readversion () ...
```

```
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html"); //membuka url  
repeat //loop yang berlangsung sampai akhir file url  
until urleof();  
s=urlgetline(); //membaca baris teks  
k=strfind(s,"Version",1); //mencari substring"Version". jika ditemukan akan disimpan di  
if k>0 then substring(s,k,strfind(s,<,k)-1), break; endif; //berhenti sebelum <  
end;  
urlclose();  
endfunction
```

```
> readversion
```

Version 2022-05-18

```
>function readdataurl () ...
```

```
urlopen("https://kumparan.com/berita-terkini/3-contoh-soal-desil-data-tunggal-beserta-ku  
repeat  
until urleof();  
s=urlgetline();  
k=strfind(s,"Tentukan persentil",1);  
if k>0 then substring(s,k,strfind(s,".",k)-1), break; endif;  
end;  
urlclose();  
endfunction
```

Selanjutnya kita mencoba dengan cara yang sama untuk mengambil soal dari website yang ada di internet

```
>readdataurl
```

## **Sub Topik 5: Perhitungan terkait analisis data statistika deskriptif**

Rata-rata, simpangan baku, jangkauan, modus, ukuran data, varians dan median.

### **Analisis data statistika deskriptif**

Statistika deskriptif adalah bidang ilmu statistika yang mempelajari cara-cara untuk pengumpulan, penyusunan, dan penyajian data sehingga memberikan informasi yang berguna. Perlu diketahui juga bahwa statistika deskriptif memberikan informasi hanya mengenai data yang dipunyai dan sama sekali tidak menarik inferensi atau kesimpulan apapun tentang gugus data induknya yang lebih besar.

Dalam praktiknya, analisis data statistika deskriptif bisa dilakukan dengan menerapkan sejumlah metode statistik, seperti :

#### **1. Mencari rata rata/mean**

Metode pertama yang digunakan untuk melakukan analisis statistika adalah mean atau sering disebut rata-rata. Saat akan menghitung rata-rata, kita bisa melakukan dengan cara menambahkan daftar angka kemudian membagi angka tersebut dengan jumlah item dalam daftar. Metode ini memungkinkan penentuan tren keseluruhan dari kumpulan data dan mampu mendapatkan tampilan data yang cepat dan ringkas. Manfaat dari metode ini juga termasuk perhitungan yang sederhana dan cepat.

### a. Rata-rata hitung data tunggal

---

Misalkan

$$x_1, x_2, x_3, \dots, x_n$$

adalah data yang dikumpulkan dari suatu sampel atau populasi maka rata-rata hitung untuk sampel disimbolkan dengan

$$\bar{x}$$

dan rata-rata hitung untuk populasi disimbolkan dengan

$$\mu$$

Sehingga, untuk mencari rata-rata hitung data tunggal terdapat 2 jenis rumus sebagai berikut :

1. Rata-rata hitung sampel

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

2. Rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^n X_i}{n}$$

Untuk menghitung rata-rata data tunggal dengan EMT, kita dapat menggunakan sintaks

> mean ([data])

Contoh Soal:

1. Diketahui data usia(dalam tahun) penduduk suatu daerah adalah sebagai berikut:

60,70,66,75,77,68,45,30,15,71,69,84,13

hitunglah rata-rata usia penduduk tersebut.

Jawab :

```
>mean([60,70,66,75,77,68,45,30,15,71,69,84,13])
```

57.1538461538

Jadi, rata rata data tersebut adalah 57.1538461538

2. Nilai ulangan matematika dari 10 siswa adalah 80, 88, 70, 60, 90, 75, 92, 78, 67, 90. Tentukan rata-rata dari data tersebut!

Jawab :

```
>mean([80, 88, 70, 60, 90, 75, 92, 78, 67, 90])
```

79

Jadi, rata-rata dari data tersebut yaitu 79

## b. Rata-rata data tabel distribusi

---

Jika diberikan data

$$x_1, x_2, \dots, x_n$$

yang memiliki frekuensi berturut-turut

$$f_1, f_2, \dots, f_n$$

maka, rataan hitung dari data yang disajikan dalam daftar distribusi tersebut ditentukan dengan 2 jenis rumus sebagai berikut :

1. Rata-rata hitung sampel

Untuk rata-rata hitung sampel,

$$\bar{x} = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i}$$

2. Rata-rata hitung populasi

Untuk rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i}$$

Cara diatas adalah beberapa perhitungan untuk mencari rata-rata data tabel distribusi menggunakan metode yang ada dalam statistika. Dengan menggunakan EMT kita juga bisa menghitung rata-rata data tabel distribusi dengan mudah, yaitu dengan cara berikut:

1. Mendeskripsikan data dan frekuensi

2. Menghitung rata-rata menggunakan perintah berikut :

```
> mean(data,frekuensi)
```

Contoh soal:

Diberikan data berat badan siswa kelas V SD yang memiliki jumlah siswa sebanyak 35 orang anak. anak dengan berat 30kg terdapat 5 orang, anak dengan berat 35kg terdapat 11 orang, anak dengan berat 40kg terdapat 4 orang, anak dengan berat 38kg terdapat 7 orang, anak dengan berat 44kg terdapat 7 orang, dan anak dengan berat 50kg terdapat 1 orang. Tentukan rata-rata berat siswa kelas V SD tersebut!

Jawab :

```
>printfile("tabel berat badan kelas V SD.dat",7); //meringkas informasi pada soal dengan m
```

Berat Badan (Kg)	Frekuensi
30	5
35	11
38	7
40	4
44	7
50	1

```
>data=[30,35,38,40,44,50];//mendefinisikan data sebagai berat siswa dalam satuan kilogram
```

```
[30, 35, 38, 40, 44, 50]
```

```
>frekuensi=[5,11,7,4,7,1];//mendefinisikan frekuensi sebagai banyak siswa
```

```
[5, 11, 7, 4, 7, 1]
```

```
>mean(data,frekuensi) //menghitung rata-rata
```

```
37.6857142857
```

Jadi, rata-rata berat badan siswa SD kelas V adalah 37.6857142857 **c. Rata-rata hitung data kelompok**

---

Misalkan suatu data kelompok terdiri dari n kelas dengan nilai tengah masing-masing kelas secara berturut-turut adalah

$$t_1, t_2, \dots, t_n$$

dan masing-masing frekuensinya adalah

$$f_1, f_2, \dots, f_n$$

Untuk mencari rata rata hitung data tersebut terdapat 2 jenis rumus sebagai berikut :

1. Rata-rata hitung sampel  
untuk rata-rata hitung sampel,

$$\bar{x} = \frac{\sum_{i=1}^n t_i f_i}{\sum_{i=1}^n f_i}$$

2. Rata-rata hitung populasi  
untuk rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^n t_i f_i}{\sum_{i=1}^n f_i}$$

Untuk menghitung rata-rata data kelompok di EMT dapat dilakukan dengan langkah berikut :

1. Menentukan tepi bawah kelas( $T_b$ ), panjang kelas( $P$ ), dan tepi atas kelas( $T_a$ ) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0,5$$

Keterangan :

a = batas bawah kelas

b = batas atas kelas

2. Membuat data menjadi bentuk tabel, dengan perintah

> r= tepi bawah terkecil : panjang kelas : tepi atas terbesar;

f=[frekuensi];

>T:r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f';

writetable(T, labc=["tepi bawah", "tapi atas", "frekuensi"]);

3. Menghitung nilai tengah kelas, dengan perintah

>T[,1]+T[,2]/2

4. Mengubah baris menjadi kolom

>t=fold(r,[0.5,0.5])

5. Menghitung rata-rata, dengan perintah

>mean(t,f)

Contoh soal :

1. Disajikan data kelompok seperti berikut :

```
>printfile("Tabel rata-rata data kelompok.dat",7)
```

Kelas	Frekuensi
31-40	3
41-50	5
51-60	10
61-70	11
71-80	8
81-90	3

```
>31-0.5 //Tepi bawah terkecil
```

30.5

```
>(40-31)+1 //Panjang kelas
```

10

```
>90+0.5 //Tepi atas kelas
```

90.5

```
>r=30.5:10:90.5; f=[3, 5, 10, 11, 8, 3];
>T:=r[1:6]' | r[2:7]' | f' ; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

tepi bawah	tepi atas	frekuensi
30.5	40.5	3
40.5	50.5	5
50.5	60.5	10
60.5	70.5	11
70.5	80.5	8
80.5	90.5	3

```
>t=(T[,1]+T[,2])/2 //menghitung nilai tengah kelas
```

35.5  
45.5  
55.5  
65.5  
75.5  
85.5

```
>t=fold(r,[0.5,0.5]) // mengubah tampilan data kolom menjadi baris dan sebaliknya
```

[35.5, 45.5, 55.5, 65.5, 75.5, 85.5]

```
>mean(t,f)
```

61.75

Jadi, rata-rata data kelompok tersebut adalah 61,75

2. Diberikan data kelompok berikut yang mewakili jumlah jam belajar per minggu dari sekelompok siswa :

```
>printfile("Tabel data kelompok conso 2.dat",5)
```

Kelompok	Frekuensi
10-14	5
15-19	8
20-24	12
25-29	6

Hitunglah rata-rata jumlah jam belajar per minggu dari data kelompok tersebut!

Jawab :

```
>10-0.5 //tepi bawah terkecil
```

9.5

```
>(14-10)+1 //panjang kelas
```

5

```
>29+0.5 //tepi atas terbesar
```

29.5

```
>r=9.5:5:29.5; f=[5, 8, 12, 6];
>T:=r[1:4]' | r[2:5]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

tepi bawah	tepi atas	frekuensi
9.5	14.5	5
14.5	19.5	8
19.5	24.5	12
24.5	29.5	6

```
>t=(T[,1]+T[,2])/2 // menghitung nilai tengah kelas
```

12  
17  
22  
27

```
>t=fold(r,[0.5,0.5]) // mengubah tampilan data kolom menjadi baris dan sebaliknya
```

[12, 17, 22, 27]

```
>mean(t,f)
```

20.064516129

Jadi, rata-rata data kelompok tersebut yaitu 20.064516129

## 2. Mencari median

---

Median ( $M_e$ ) adalah nilai tengah dari suatu data yang telah disusun dari data terkecil sampai data terbesar atau sebaliknya. Selain sebagai ukuran pemusatan data, median juga dijadikan sebagai ukuran letak data dan dikenal sebagai kuartil 2 (Q2). Rumus perhitungan median dibedakan untuk data tak berkelompok dan data berkelompok.

### a. Median data tunggal

---

Median data tunggal adalah mengurutkan data berdasarkan nilainya, misalkan data yang telah terurut dari data terkecil ke data terbesar adalah

$$x_1, x_2, \dots, x_n$$

untuk menentukan letak median dengan menggunakan rumus :

1. Jika jumlah suatu data(n) berjumlah ganjil maka nilai mediannya adalah sama dengan data yang memiliki nilai di urutan paling tengah yang memiliki nomor urut k, dimana untuk menentukan nilai k dapat dihitung menggunakan rumus:

$$k = \frac{n+1}{2}$$

2. Jika jumlah suatu data (n) berjumlah genap, maka untuk menghitung mediannya dengan menggunakan rumus :

$$k = \frac{n}{2}$$

$$\text{Median} = \frac{1}{2}(x_k + x_{k+1})$$

Diatas adalah rumus untuk mencari median secara statistika. Dengan menggunakan EMT kita bisa menentukan median dengan menggunakan perintah

> median([data])

perintah tersebut dapat berjalan dengan baik apabila data sudah diurutkan terlebih dahulu dari data terkecil hingga terbesar.

Contoh soal :

Diketahui data hasil tes SKD calon PNS adalah sebagai berikut :

487, 300, 450, 500, 521, 440

Tentukan nilai median dari data tersebut!

Jawab :

```
>data=[487, 300, 450, 500, 521, 440]; //mendeskripsikan data  
>urutan=sort(data) //mengurutkan data
```

```
[300, 440, 450, 487, 500, 521]
```

```
>median([urutan])
```

468.5

Jadi, nilai median dari data hasil tes SKD adalah 468.5

## b. Median data kelompok

Menghitung median data kelompok dapat menggunakan rumus di bawah ini :

$$M_e = Tb + p \frac{\frac{1}{2}n - F}{f}$$

Keterangan:

T<sub>b</sub> = tepi bawah kelas median, ialah kelas dimana median terletak

p = panjang kelas median

n = ukuran sampel / banyak data

F = jumlah semua frekuensi dengan tanda kelas lebih kecil dari tanda kelas median.

f = frekuensi kelas median

Untuk menghitung median data berkelompok di EMT, dapat dilakukan dengan cara berikut:

1. Menentukan tepi bawah kelas (T<sub>b</sub>), panjang kelas (P), dan tepi atas kelas (T<sub>a</sub>) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0,5$$

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];  
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"]);

3. Mendeskripsikan batas bawah kelas median, panjang kelas median, banyak data, jumlah frekuensi sebelum kelas median, frekuensi median

> Tb=(tepi bawah kelas median), p=(panjang kelas median), n=(banyak data), F=(jumlah frekuensi sebelum kelas median), f=(frekuensi kelas median)

4. Menghitung median data dengan perintah:

> Tb+p\*(1/2\*n-F)/f

Contoh soal :

Berikut adalah data hasil dari pengukuran berat badan 20 siswa SD kelas V. Dari ke 20 siswa, siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 4 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 3 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 2 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 3 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 3 orang. Tentukan median dari

data hasil pengukuran berat badan 20 siswa di SD tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

```
>21-0.5 // menentukan tepi bawah kelas terkecil
```

20.5

```
> (26-21)+1 // menentukan panjang kelas
```

6

```
> 56+0.5 // tepi atas kelas terbesar
```

56.5

```
>r=20.5:6:56.5; f=[5, 4, 3, 2, 3, 3];
>T :=r[1:6]' | r[2:7]' | f' ; writetable(T, labc=["Tb", "Ta", "frekuensi"])
```

Tb	Ta	frekuensi
20.5	26.5	5
26.5	32.5	4
32.5	38.5	3
38.5	44.5	2
44.5	50.5	3
50.5	56.5	3

```
>Tb=32.5, p=6, n=20, F=9, f=3
```

32.5  
6  
20  
9  
3

```
>Tb+p*(1/2*n-F)/f
```

34.5

Jadi, median dari data hasil pengukuran berat badan 20 siswa SD kelas V adalah 34.5

### 3. Mencari Modus

---

Modus adalah area fokus dalam analisis statistika deskriptif yang termasuk dalam ukuran pusat data. Ini adalah nilai yang paling sering muncul dalam kumpulan data atau nilai yang memiliki frekuensi tertinggi dalam distribusi data. Modus dapat dibagi menjadi dua jenis, yaitu modus untuk data tunggal dan modus untuk data kelompok.

#### a.Modus untuk data tunggal:

---

Menentukan modus untuk data tunggal cukup sederhana. Pertama, data diurutkan dari nilai terkecil ke terbesar sehingga data dengan nilai yang sama berdekatan satu sama lain. Selanjutnya, frekuensi masing-masing data dihitung, dan data yang memiliki frekuensi tertinggi dipilih sebagai modus.

#### b.Modus untuk data kelompok

---

Berikut rumus untuk mencari modus data kelompok :

$$M_o = Tb + \frac{d_1}{d_1 + d_2} c$$

Keterangan :

Tb = Tepi bawah

d1 = selisih f modus dengan f sebelumnya

d2 = selisih f modus dengan f sesudahnya

c = Panjang kelas

Untuk menghitung modus data berkelompok di EMT, dapat dilakukan dengan cara berikut:

1. Menentukan tepi bawah kelas (Tb), panjang kelas (P), dan tepi atas kelas (Ta) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0,5$$

dimana a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; v=[frekuensi];  
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"]);

3. Mendeskripsikan tepi bawah kelas modus, panjang kelas modus, selisih frekuensi modus dengan frekuensi sebelumnya, selisih frekuensi modus dengan frekuensi sesudahnya

> Tb=(tepi bawah kelas modus), p=(panjang kelas modus), d1=(selisih frekuensi modus dengan frekuensi sebelumnya), d2=(selisih frekuensi dengan frekuensi sesudahnya)

4. Menghitung modus dengan perintah:

> Tb+p\*d1/(d1+d2)

Contoh soal

Diketahui sebuah data kelompok sebagai berikut :

```
>printfile("Tabel modus data kelompok.dat",8)
```

Kelas	Frekuensi
20-29	3
30-39	7
40-49	8
50-59	12
60-69	9
70-79	6
80-89	5

Berapakah modus dari data tersebut?

```
>20-0.5 //menentukan tepi bawah kelas
```

19.5

```
>(29-20)+1 //menentukan panjang kelas
```

```
>89+0.5 //menentukan tepi atas
```

89.5

```
>r=19.5:10:89.5; f=[3, 7, 8, 12, 9, 6, 5];
>T:=r[1:7]' | r[2:8]' | f'; writetable(T,labc=["Tb", "Ta", "frekuensi"])
```

Tb	Ta	frekuensi
19.5	29.5	3
29.5	39.5	7
39.5	49.5	8
49.5	59.5	12
59.5	69.5	9
69.5	79.5	6
79.5	89.5	5

Berdasarkan tabel di atas, modus berada pada kelas 49.5-59.5

```
>Tb=49.5, p=10, d1=12-8, d2=12-9
```

49.5  
10  
4  
3

```
>Tb+p*d1/ (d1+d2)
```

55.2142857143

Jadi, modus dari data kelompok di atas adalah 55.2142857143

#### 4. Mencari varians/ragam

---

Varians digunakan untuk mengetahui bagaimana sebaran data terhadap mean atau nilai rata-rata. Sederhananya, varians adalah ukuran statistik jauh dekatnya penyebaran data dari nilai rata-ratanya. Dalam mencari ragam dapat dikelompokkan menjadi 2 jenis yaitu sebagai berikut :

##### a. Varians data tunggal

---

Rumus untuk varians data tunggal berikut :

1) Untuk populasi

$$\sigma^2 = \frac{\sum_{i=1}^n (x - \mu)^2}{n}$$

## 2) Untuk sampel

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Pada EMT, untuk menemukan suatu Ragam data tunggal dapat menggunakan perintah berikut:

```
> mean(dev^2)
```

Contoh soal

Hitunglah nilai varians dari data sampel nilai siswa: 9, 10, 6, 7!

```
>data=[9, 10, 6, 7]; //mendefinisikan data  
>urut=sort(data) //mengurutkan data
```

```
[6, 7, 9, 10]
```

```
>xbar=mean(urut) //menghitung rata rata dari data
```

```
8
```

```
>dev= urut-xbar
```

```
[-2, -1, 1, 2]
```

```
>varians=mean(dev^2) //menghitung varians
```

```
2.5
```

Jadi, varians dari data sampel tersebut adalah 2.5

## b. Varians data kelompok

Rumus untuk varians data kelompok sebagai berikut :

1) Untuk populasi

$$\sigma^2 = \frac{\sum_{i=1}^n f_i(x_i - \mu)^2}{\sum_{i=1}^n f_i}$$

2) Untuk sampel

$$S^2 = \frac{\sum_{i=1}^n f_i(x_i - \bar{x})^2}{\sum_{i=1}^n f_i - 1}$$

Pada EMT, untuk menemukan Ragam data berkelompok dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas ( $T_b$ ), panjang kelas ( $P$ ), dan tepi atas kelas ( $T_a$ ) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan  $a$  = batas bawah kelas dan  $b$  = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

```
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])
```

3. Menghitung Ragam dengan perintah

```
> (T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]);m=mean(t,f);
> sum(f*(t-m)^2)/sum(f) //untuk populasi
> sum(f*(t-m)^2)/(sum(f)-1) //untuk sampel
```

Contoh soal

Tentukan varians data sampel dari tabel berikut :

```
>printfile("Tabel data kelompok varians.dat",7)
```

Nilai	Frekuensi
63–67	3
68–72	2
73–77	7
78–82	3
83–87	4
88–92	1

```
>63-0.5 //tapi bawah terkecil
```

62.5

```
>(67-63)+1 //panjang kelas
```

5

```
>92+0.5 //tepi atas terbesar
```

92.5

```
>r=62.5:5:92.5; f=[3,2,7,3,4,1];
>T:=r[1:6]' | r[2:7]' | f'; writetable(T, labc=["tepi bawah", "tepi atas", "frekuensi"])
```

tepi bawah	tepi atas	frekuensi
62.5	67.5	3
67.5	72.5	2
72.5	77.5	7
77.5	82.5	3
82.5	87.5	4
87.5	92.5	1

```
>(T[,1]+T[,2])/2; t=fold(r,[0.5,0.5])
```

[65, 70, 75, 80, 85, 90]

```
>m=mean(t,f)
```

76.5

```
>sum(f*(t-m)^2)/(sum(f)-1)
```

52.8947368421

Jadi, varians dari data kelompok dari tabel di atas adalah 52.8947368421

```
>
```

## 5. Mencari Simpangan Baku

---

Standar Deviasi atau simpangan baku adalah akar dari ragam/varians. Untuk menentukan nilai standar deviasi, caranya:

$$\sigma = \sqrt{\sigma^2}$$

atau

$$S = \sqrt{S^2}$$

### a. Simpangan baku data tunggal

---

Untuk data tunggal, simpangan baku populasi atau sampel dapat dirumuskan sebagai berikut:

1) Untuk populasi

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x - \mu)^2}{n}}$$

2) Untuk sampel

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

Pada EMT, untuk menemukan suatu Ragam data tunggal dapat menggunakan perintah berikut:

```
> sqrt(mean(dev^2))
```

Contoh soal :

1. Simpangan baku untuk data 70,80,40,25,65,87,97,59,24,77,45 adalah

Jawab :

```
>data=[70,80,40,25,65,87,97,59,24,77,45];  
>urut=sort(data)
```

```
[24, 25, 40, 45, 59, 65, 70, 77, 80, 87, 97]
```

```
>x=mean(urut)
```

```
60.8181818182
```

```
>dev=urut-x
```

```
[-36.8182, -35.8182, -20.8182, -15.8182, -1.81818, 4.18182,  
9.18182, 16.1818, 19.1818, 26.1818, 36.1818]
```

```
>varians=mean(dev^2)
```

```
550.148760331
```

```
>simpanganbaku= sqrt(varians)
```

```
23.4552501656
```

Jadi, simpang baku data tersebut adalah 23.4552501656

## b. Simpangan baku data kelompok

Untuk data berkelompok dapat dirumuskan seperti berikut:

1) Untuk populasi

$$\sigma = \sqrt{\frac{\sum_{i=1}^n f_i(x_i - \mu)^2}{\sum_{i=1}^n f_i}}$$

## 2) Untuk sampel

$$S = \sqrt{\frac{\sum_{i=1}^n f_i (x_i - \bar{x})^2}{\sum_{i=1}^n f_i - 1}}$$

Pada EMT, untuk menemukan Ragam data berkelompok dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas (T<sub>b</sub>), panjang kelas (P), dan tepi atas kelas (T<sub>a</sub>) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

```
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])
```

3. Menghitung Ragam dengan perintah

```
> (T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]);m=mean(t,f);
> sqrt(sum(f*(t-m)^2)/sum(f)) // untuk populasi
> sqrt(sum(f*(t-m)^2)/(sum(f)-1)) // untuk sampel
```

Contoh soal :

Simpangan baku dari tabel dibawah ini adalah

```
>printfile("Tabel simpangan baku data kelompok.dat",7)
```

Interval nilai	Frekuensi
41–45	10
46–50	12
51–55	18
56–60	34
61–65	20
66–70	6

```
>41-0.5 //tepi bawah terkecil
```

40.5

```
>(45-41)+1 //panjang kelas
```

5

```
>70+0.5 //tepi atas terbesar
```

70.5

```
>r=40.5:5:70.5; f=[10,12,18,34,20,6];
>T:=r[1:6]' | r[2:7]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

tepi bawah	tepi atas	frekuensi
40.5	45.5	10
45.5	50.5	12
50.5	55.5	18
55.5	60.5	34
60.5	65.5	20
65.5	70.5	6

```
>(T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]); m=mean(t,f);
```

karena data tersebut merupakan data sampel, maka menggunakan rumus berikut

```
>sqrt(sum(f*(t-m)^2)/(sum(f)-1))
```

6.81649810861

Jadi, simpangan baku data kelompok tersebut adalah 6.81649810861

## 6. Mencari Jangkauan/Range

---

Jangkauan, atau biasa disebut range, merupakan perbedaan antara nilai data tertinggi dan nilai data terendah dalam suatu set data. Metode pencarian jangkauan berbeda antara data tunggal dan data kelompok.

### a. Jangkauan/Range Data Tunggal

---

Bila ada sekumpulan data tunggal terurut dari yang terkecil sampai terbesar adalah

$$x_1, x_2, \dots, x_n$$

maka jangkauannya adalah:

$$\text{Jangkauan} = x_n - x_1$$

Untuk menemukan jangkauan data tunggal di EMT dapat menggunakan perintah berikut:

```
>x=[data]; max(x)-min(x)
```

Contoh soal

Jangkauan dari data 30,60,87,55,87,98,22,75,81,70,69,84,75 adalah...

```
>x=[30, 60, 87, 55, 87, 98, 22, 75, 81, 70, 69, 84, 75]; max(x) - min(x)
```

Jadi, jangkauan dari data tersebut adalah 76

## b. Jangkauan data kelompok

Jangkauan pada data berkelompok adalah selisih antara batas atas dari kelas tertinggi dengan batas bawah dari kelas terendah.

Pada EMT, untuk menemukan jangkauan dari data berkelompok dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas ( $T_b$ ), panjang kelas ( $P$ ), dan tepi atas kelas ( $T_a$ ) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan  $a$  = batas bawah kelas dan  $b$  = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

```
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas] | r[2:jumlah kelas + 1] | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])
```

3. Menghitung jangkauan data berkelompok

```
> max(transpose(T[,2]))-min(transpose(T[,1]))
```

Contoh soal :

Berikut adalah data hasil dari pengukuran berat badan 20 siswa SD kelas V. Dari ke 20 siswa,siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 4 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 3 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 2 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 3 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 3 orang. Tentukan jangkauan dari

data hasil pengukuran berat badan 20 siswa di SD tersebut!

Jawab :

```
>printfile("Tabel jangkauan data kelompok.dat",7) //menyederhanakan informasi
```

Interval nilai	Frekuensi
21-26	5
27-32	4
33-38	3
39-44	2
45-50	3
51-56	3

```
>21-0.5 //tepi bawah terkecil
```

20.5

```
>(26-21)+1 //panjang kelas
```

```
>56+0.5 //tepi atas terbesar
```

56.5

```
>r=20.5:6:56.5; f=[5,4,3,2,3,3];
>T:=r[1:6]' | r[2:7]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])
```

tepi bawah	tepi atas	frekuensi
20.5	26.5	5
26.5	32.5	4
32.5	38.5	3
38.5	44.5	2
44.5	50.5	3
50.5	56.5	3

```
>max(transpose(T[,2]))-min(transpose(T[,1]))
```

36

Jadi, Jangkauan dari data kelompok tersebut adalah 36

## 7. Menentukan ukuran letak

---

Ukuran letak merupakan ukuran untuk melihat dimana letak salah satu data dari sekumpulan banyak data yang ada. Yang termasuk ukuran ukuran letak antara lain adalah kuartil(Q), desil(D) dan persentil(P). Dalam menentukan ke-3 nya yang harus diingat adalah mengurutkan distribusi data dari yang terkecil sampai terbesar

### 1. Kuartil

Dalam EMT untuk menghitung kuartil bisa dilakukan dengan perintah

```
>quartiles(data)
```

perintah tersebut akan menghasilkan nilai Q1, Q2, Q3, nilai minimum dan nilai maksimum dari suatu data

### 2. Desil

Dalam EMT untuk menghitung desil bisa dilakukan dengan perintah

```
>quantile(data)
```

### 3. Persentil

Dalam EMT untuk menghitung persentil bisa dilakukan dengan perintah

```
>quantile(data)
```

perintah ">quantile(data)" dapat digunakan untuk menentukan desil dan persentil perbedaananya tergantung pada nilai dari pembaginya

Contoh soal

1. Tentukan Q1,Q2 dan Q3 dari data : 7,3,8,5,9,4,8,3,10,2,7,6,8,7,2,6,9.

```
>data=[7,3,8,5,9,4,8,3,10,2,7,6,8,7,2,6,9];
>urut=sort(data)
```

[2, 2, 3, 3, 4, 5, 6, 6, 7, 7, 7, 8, 8, 8, 9, 9, 10]

```
>quartiles(urut)
```

```
[2, 3.5, 7, 8, 10]
```

dari hasil di atas diperoleh nilai sebagai berikut :

Nilai minimal data = 2

Q1=3.5

Q2=7

Q3=8

Nilai maksimal data = 10

2. Tentukan D8 dari data : 6,3,8,9,5,9,9,7,5,7,4,5,8,3,7,6

```
>data=[6,3,8,9,5,9,9,7,5,7,4,5,8,3,7,6];
>urut=sort(data)
```

```
[3, 3, 4, 5, 5, 5, 6, 6, 7, 7, 7, 8, 8, 9, 9, 9]
```

```
>quantile(urut,0.8) //nilai 0.8 diapatkan karena kita akan mencari D8
```

8

Jadi, nilai dari D8 berdasarkan perhitungan di atas adalah 8

3. Tentukan persentil ke-65 dari data : 6,5,8,7,9,4,5,8,4,7,8,5,8,4,5

```
>data=[6,5,8,7,9,4,5,8,4,7,8,5,8,4,5];
>urut=sort(data)
```

```
[4, 4, 4, 5, 5, 5, 5, 6, 7, 7, 8, 8, 8, 8, 8, 9]
```

```
>quantile(urut,65%)
```

7.1

## Sub Topik 6: Menggambar Grafik Statistika

### Diagram Kotak

Diagram kotak atau box plot merupakan ringkasan distribusi sampel yang disajikan secara grafis yang bisa menggambarkan bentuk distribusi data (skewness), ukuran tendensi sentral dan ukuran penyebaran (keragaman) data pengamatan. Diagram kotak sering digunakan ketika jumlah distribusi data perlu dibandingkan. Diagram kotak menyajikan informasi tentang nilai-nilai inti dalam distribusi data termasuk juga pencilan. Pencilan adalah titik data yang terpaut jauh dari titik data lainnya.

Contoh:

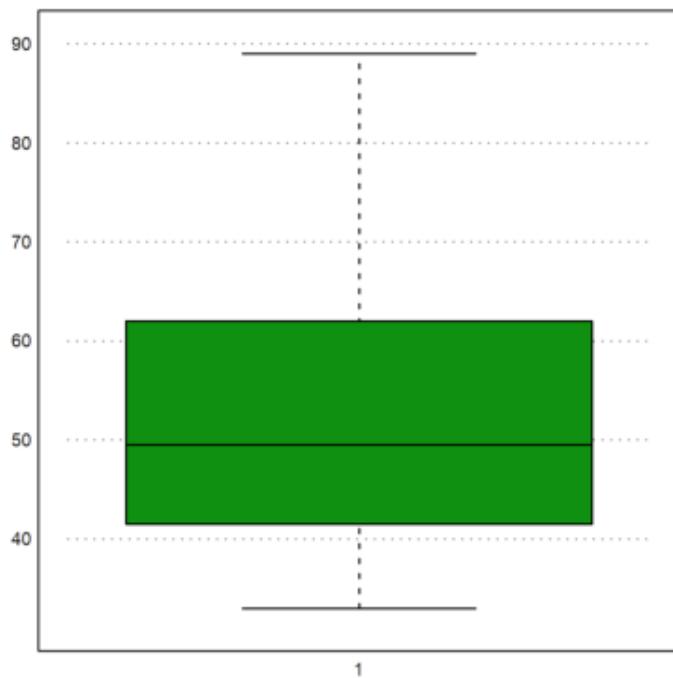
Diketahui data berat badan mahasiswa di Universitas A sebagai berikut.

```
>A=[55, 50, 33, 42, 44, 37, 63, 74, 56, 34, 51, 43, 45, 39, 64, 77, 60, 35, 53, 43, 48, 41, 65, 87, 61, 36, 54, 44, 49]
```

```
[55, 50, 33, 42, 44, 37, 63, 74, 56, 34, 51, 43, 45, 39, 64, 77, 60, 35, 53, 43, 48, 41, 65, 87, 61, 36, 54, 44, 49, 49, 41, 66, 89]
```

Buatlah diagram kotak (box plot) kemudian tuliskan interpretasinya.

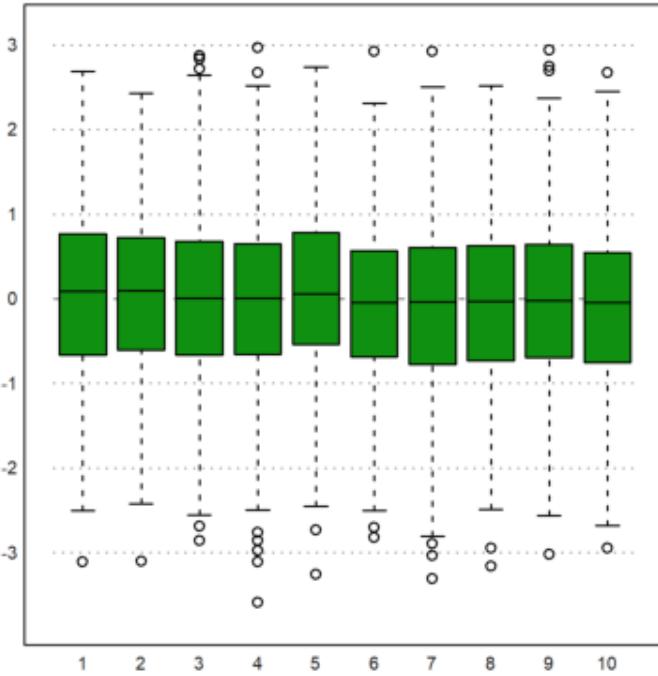
```
>boxplot (A) :
```



Dari gambar box plot berat badan mahasiswa Universitas A, sepintas kita bisa menentukan beberapa ukuran statistik, meskipun tidak persis sekali. Nilai statistik pada badan boxplot berkisar pada: Nilai Minimum = 33 , Q1 = 41.5 , Median (Q2) = 49.5 , Q3 = 62 , Nilai Maksimum = 89 . Sebaran data tidak simetris, melainkan menjulur ke arah kanan (positively skewness). Karena nilai jarak Q1 dengan Q2 lebih pendek dari jarak Q2 dengan Q3, maka data lebih terpusat di kiri. Akan tetapi data tersebut tergolong cenderung mesokurtik karena jarak IQR dengan panjang hampir sama, dengan data berpusat di angka 49.5

Adapun contoh perbandingan 10 simulasi 500 nilai terdistribusi normal menggunakan box plot dan terdapat penciran sebagai berikut.

```
> p=normal(10,500); boxplot(p) :
```



pada diagram diatas, adalah membuat boxplot distribusi normal dengan rata-rata 10 dan standar deviasi 500. Boxplot adalah representasi grafis dari lokalitas, penyebaran, dan kecondongan sekelompok data numerik melalui kuartil mereka

2

## Diagram Batang

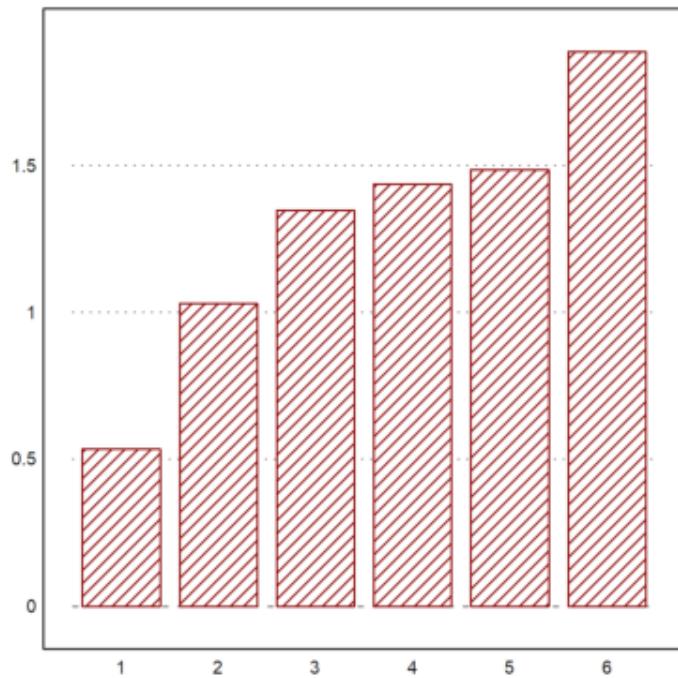
---

Diagram batang adalah representasi visual dari data yang menggunakan balok atau kolom vertikal untuk mewakili kategori, nilai atau variabel tertentu. Setiap kolom yang ada pada diagram batang memiliki frekuensi atau jumlah dalam kategori tersebut.

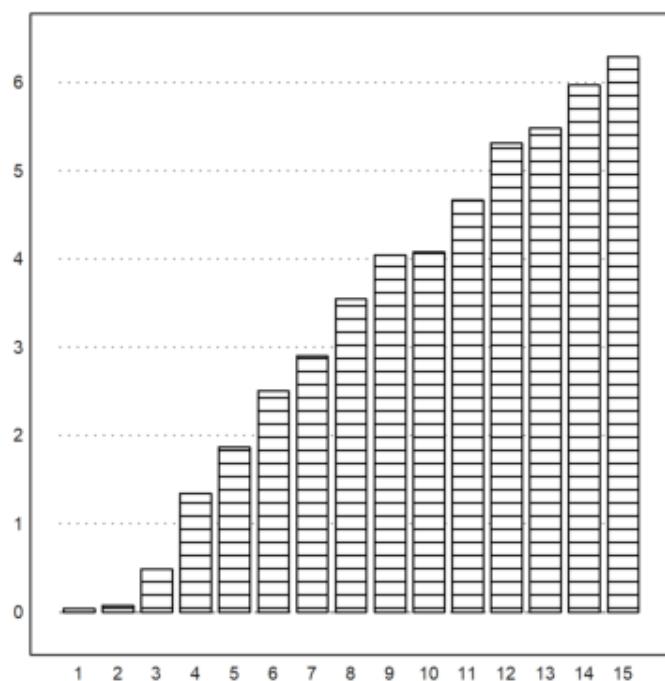
Contoh:

Kita akan membuat diagram batang secara random.

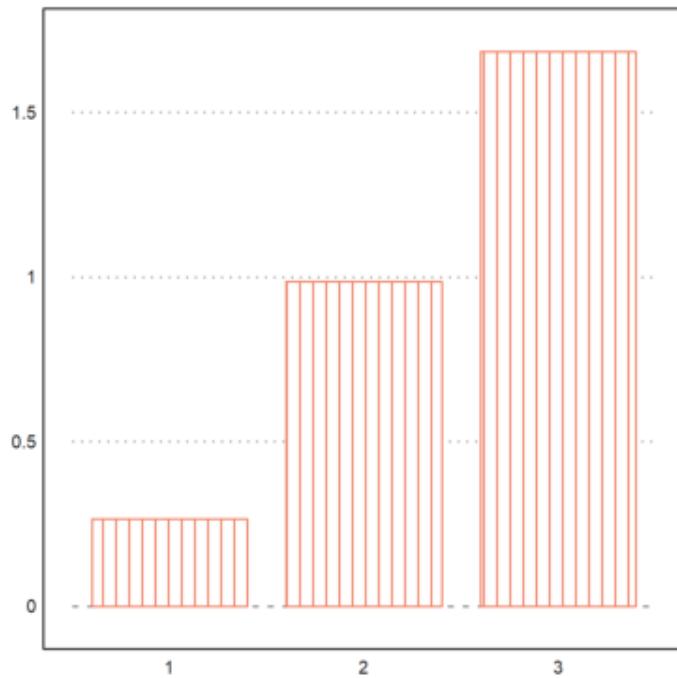
```
>columnsplot(cumsum(random(6)), style="/", color=red) :
```



```
>columnsplot(cumsum(random(15)), style="-", color=black) :
```

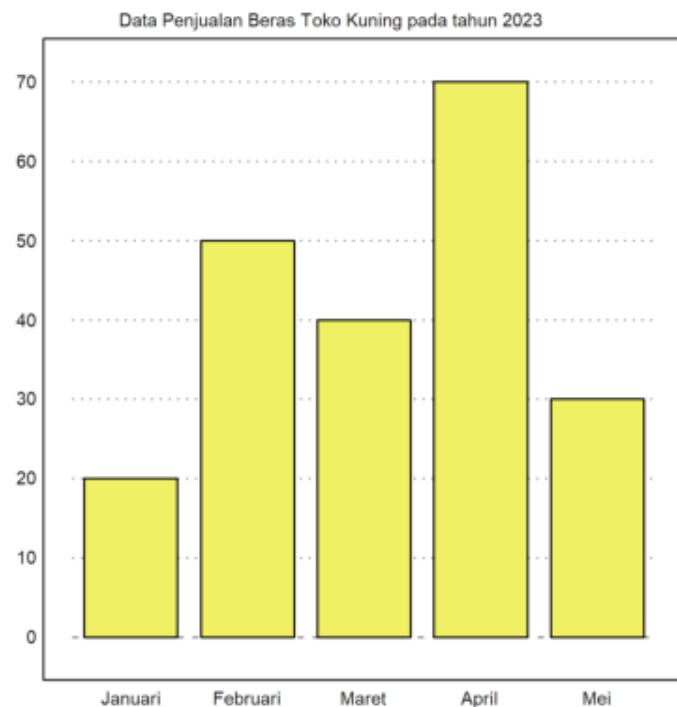


```
>columnsplot(cumsum(random(3)), style="|", color=orange) :
```



Selanjutnya kita akan mencoba membuat diagram batang penjualan yang menggunakan variabel.

```
>months=["Januari","Februari","Maret","April","Mei"];
>values=[20,50,40,70,30];
>columnspplot(values,lab=months,color=yellow);
>title("Data Penjualan Beras Toko Kuning pada tahun 2023");
```



Perintah "columnsplot(values,lab=months,color=yellow);" merupakan sintaks untuk membuat diagram batang dengan menggunakan nilai dari variabel "values", label bulan dari variabel "months", dan warna kuning. Dari diagram batang tersebut kita bisa mengetahui data penjualan toko kuning selama lima bulan pada tahun 2023 yaitu, pada bulan Januari, Februari, Maret, April, Mei. Januari terjual 20 ton beras, Februari terjual 50 ton beras, Maret terjual 40 ton beras, April terjual 70 ton beras, dan Mei terjual 30 ton beras.

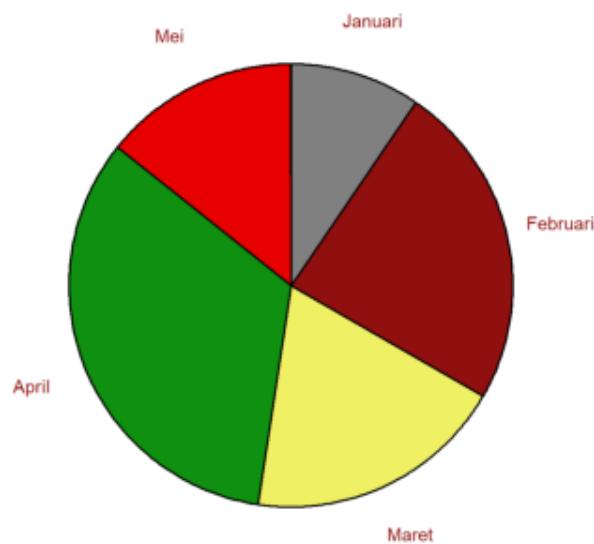
## Diagram Lingkaran

Diagram lingkaran merupakan penyajian statistik data tunggal dalam bentuk lingkaran yang dibagi menjadi beberapa juring atau sektor yang menggambarkan banyak frekuensi untuk setiap data. Diagram lingkaran tidak menampilkan informasi frekuensi dari masing-masing data secara detail.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.9,0,0)]
```

```
[5.87532e+07, 2, 15, 3, 6.54049e+07]
```

```
>i=[1,2,3,4,5]; piechart(values[i],color=CP[i],lab=months[i]):
```



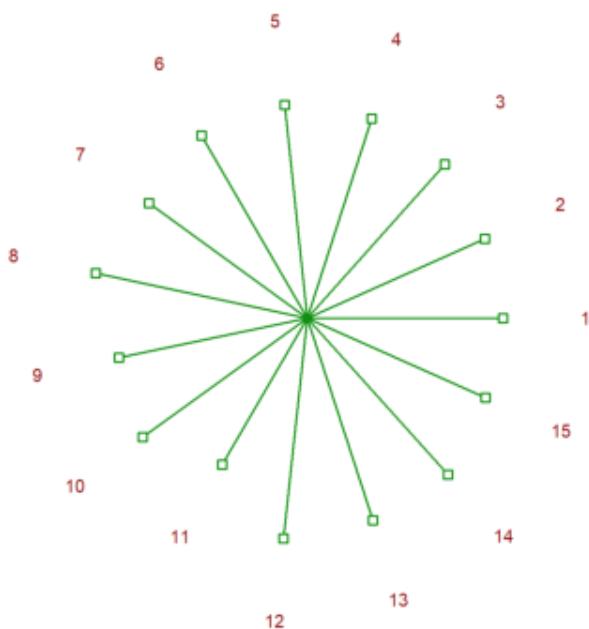
RGB adalah singkatan dari Red, Green, and Blue, dan setiap parameter mendefinisikan intensitas warna dengan nilai antara 0 dan 1. Warna pertama dalam daftar adalah warna abu-abu dengan jumlah merah, hijau, dan biru yang sama. Warna kedua merah, ketiga kuning, dan keempat hijau. Warna terakhir adalah warna merah dengan lebih banyak merah daripada hijau atau biru.

## Diagram Bintang

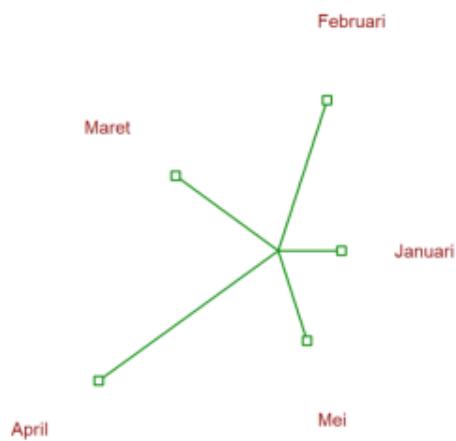
Diagram bintang, terkadang disebut diagram radar atau diagram web, adalah metode perangkat grafis yang digunakan untuk menampilkan data multivariat. Multivariat dalam pengertian ini mengacu pada memiliki banyak karakteristik untuk diamati. Variabelnya juga harus berupa nilai yang berkisar.

Diagram bintang terdiri dari rangkaian jari-jari bersudut sama, yang disebut jari-jari, dengan masing-masing jari mewakili salah satu variabel. Panjang jari-jari data sebanding dengan besaran variabel pada titik data relatif terhadap besaran maksimum variabel di seluruh titik data.

```
>starplot(normal(1,15)+16,lab=1:15,>rays) :
```



```
>starplot(values,lab=months,>rays) :
```



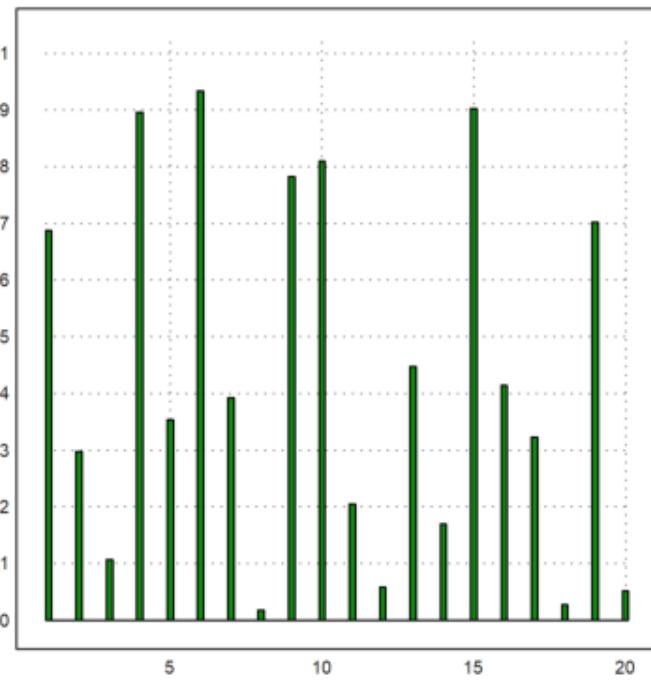
Syntax `starplot(values,lab=months,rays)` adalah perintah untuk membuat grafik bintang (star plot) dengan menggunakan nilai-nilai yang diberikan dalam vektor `values`, label sumbu yang diberikan dalam vektor `months`, dan jumlah `rays` yang menentukan jumlah garis radial yang digunakan dalam grafik

### Diagram Impuls

Impuls (impulse) adalah perubahan momentum. Contohnya adalah sebuah bola bermassa yang tengah ditendang, bola menggelinding yang dihentikan, bola jatuh yang memantul, mobil yang menabrak tembok, telur jatuh yang pecah.

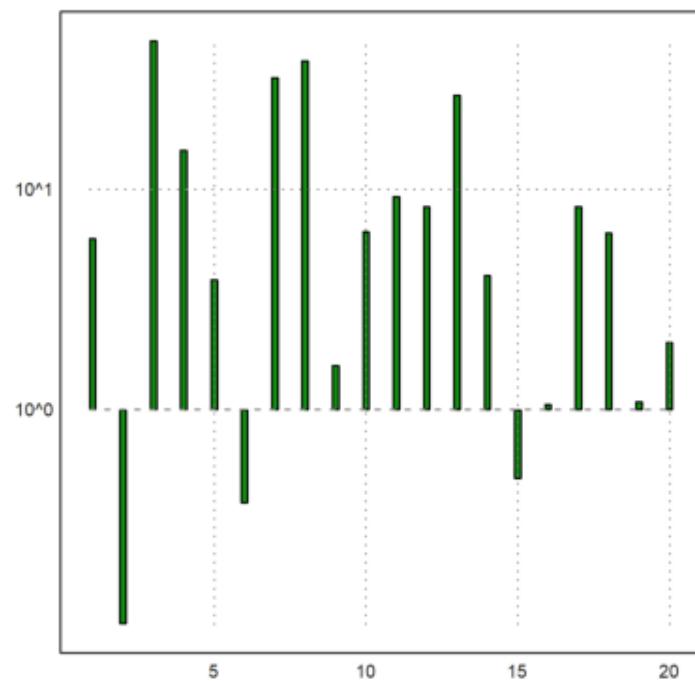
Berikut adalah plot impuls dari data acak 1 sampai 20, terdistribusi secara merata di [0,1].

```
>plot2d(makeimpulse(1:20,random(1,20)),>bar):
```



Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
> logimpulseplot(1:20,-log(random(1,20))*10):
```

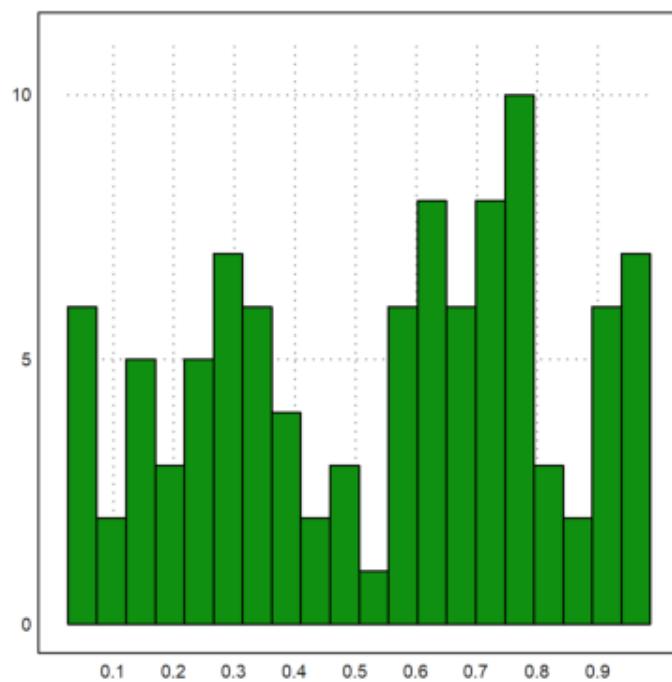


Jadi gambar grafiknya terlihat naik turun (mengalami perubahan).

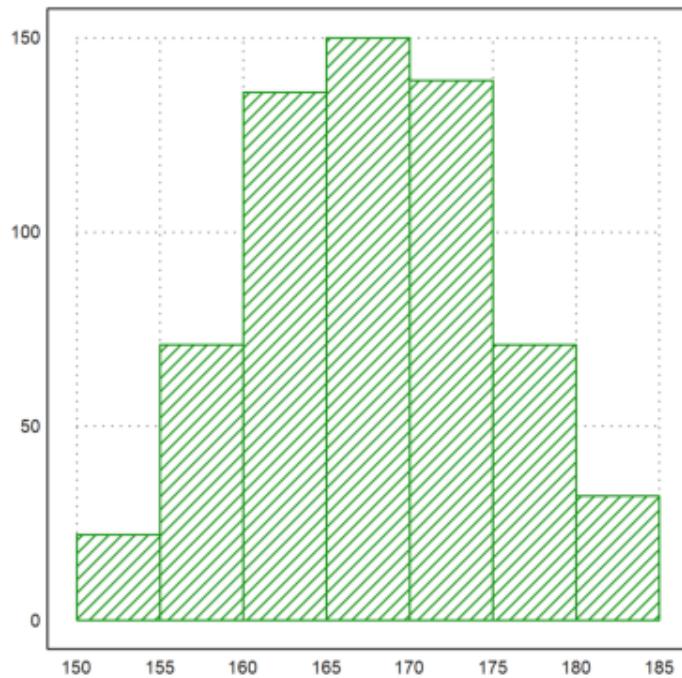
## Histogram

Histogram adalah representasi grafis (diagram) yang mengatur dan menampilkan frekuensi data sampel pada rentang tertentu. Frekuensi data yang ada pada masing-masing kelas direpresentasikan dengan bentuk grafik diagram batang atau kolom.

```
>aspect(1); plot2d(random(100),>histogram):
```



```
>r=150:5:185; v=[22,71,136,150,139,71,32];  
>plot2d(r,v,a=150,b=185,c=0,d=150,bar=1,style="/"':
```



Pola "r=150:5:185" berarti bahwa nilai r dimulai dari 150, kemudian bertambah 5 setiap kali, dan berakhir saat mencapai atau melebihi 185. Dengan pola ini, kita dapat menentukan nilai-nilai r yang sesuai.

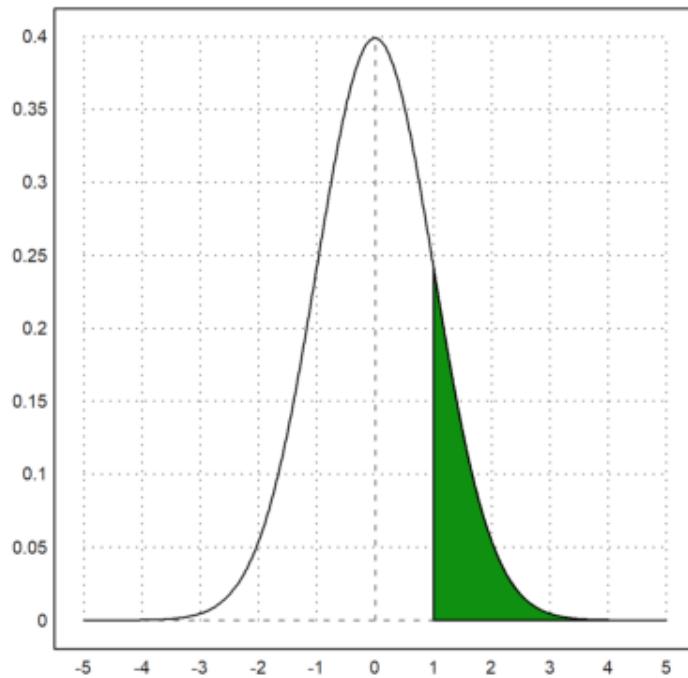
Dari data yang diperoleh dapat diketahui bahwa dari rentang kelas 150-155 memiliki frekuensi 22, rentang kelas 155-160 memiliki frekuensi 71, dan seterusnya.

### Kurva Fungsi Kerapatan Probabilitas

---

Secara teoritis kurva probabilitas populasi diwakili oleh poligon frekuensi relatif yang dimuluskan (variabel acak kontinu diperlakukan seperti variabel acak diskrit yang rapat). Karena itu fungsi dari variabel acak kontinu merupakan fungsi kepadatan probabilitas (probability density function – pdf). Pdf menggambarkan besarnya probabilitas per unit interval nilai variabel acaknya.

```
>plot2d("qnormal(x,0,1)",-5,5); ...
>plot2d("qnormal(x,0,1)",a=1,b=4,>add,>filled):
```



Perintah "plot2d("qnormal(x,0,1)",-5,5)" digunakan untuk membuat plot dari distribusi normal dengan mean 0 dan standard deviation 1 di rentang -5 hingga 5

Probabilitas variabel acak  $x$  yang terletak antara 1 dan 4 memenuhi

$P(1 < X < 4) = \text{luas daerah hijau}$

## Kurva Fungsi Distribusi Kumulatif

---

Cumulative Distribution Function (CDF) atau fungsi distribusi kumulatif adalah fungsi matematika yang digunakan untuk menghitung probabilitas variabel acak diskrit atau kontinu. CDF memberikan probabilitas bahwa variabel acak akan menghasilkan nilai kurang dari atau sama dengan nilai tertentu. Dalam hal ini, CDF dapat digunakan untuk menghitung probabilitas kumulatif dari variabel acak.

Berikut merupakan contoh kurva fungsi distribusi kumulatif kontinu:

```
>splot2d("normaldis", -3, 5) :
```

```
Function splot2d not found.  
Try list ... to find functions!  
Error in:  
splot2d("normaldis",-3,5): ...  
^
```

Dapat kita lihat dalam kurva fungsi distribusi kumulatif kontinu terdiri atas tiga bagian yaitu:

1. Bernilai 0 untuk  $x$  di bawah minimal dari daerah rentang.
2. Merupakan fungsi monoton naik pada daerah rentang.
3. Mempunyai nilai konstan 1 di atas batas maksimum daerah rentangnya.

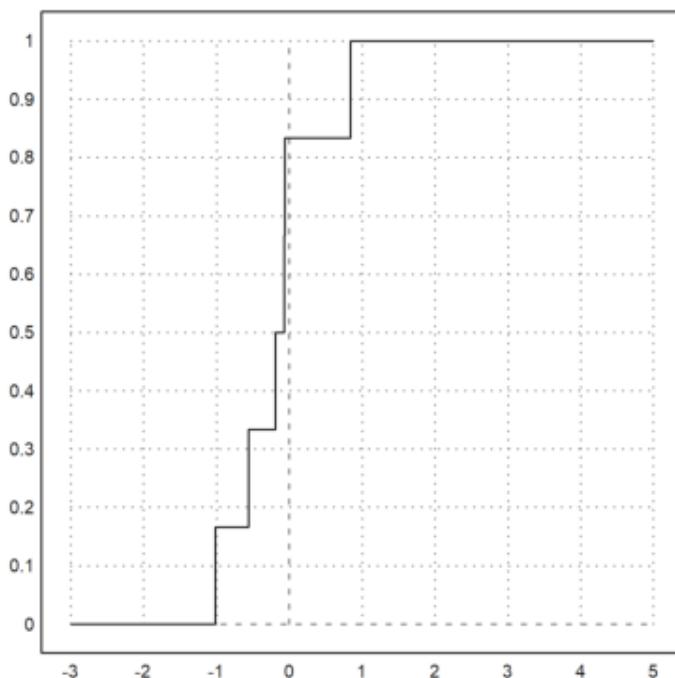
Adapun contoh kurva fungsi distribusi kumulatif diskrit sebagai berikut.

```
>x=normal(1, 6);
```

Baris kode tersebut akan menghasilkan suatu nilai acak dari distribusi normal dengan mean 1 dan deviasi standar 6, dan nilai tersebut disimpan dalam variabel x. Variabel x kemudian dapat digunakan dalam perhitungan atau analisis selanjutnya

Fungsi empdist(x,vs) membutuhkan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

```
>xs=sort(x);  
>plot2d("empdist",-3,5;xs):
```



Grafik fungsi distribusi kumulatif peubah acak diskrit merupakan fungsi tangga naik dengan nilai terendah 0 dan nilai tertinggi 1.

## Sub topik 7 : Menampilkan Tabel Frame Data

---

Cakupan Materi 1) Diagram Titik

2) Diagram Garis

3) Kurva Regresi

4) Menampilkan Tabel Data Frame

---

Diagram Titik Diagram titik atau bisa disebut Scatter Plot adalah tipe grafik yang digunakan untuk menampilkan nilai-nilai dua variabel pada sumbu horizontal dan vertikal. Setiap titik dalam diagram mewakili satu observasi atau data point. Diagram titik sangat berguna untuk menemukan pola atau hubungan antara dua variabel, serta untuk mengevaluasi distribusi data.

Dalam scatter plot, sumbu horizontal umumnya digunakan untuk variabel independen, sementara sumbu vertikal digunakan untuk variabel dependen. Dengan melihat pola penyebaran titik-titik, kita akan mendapatkan wawasan tentang apakah ada korelasi antara dua variabel dan jenis korelasi apa yang mungkin ada (positif, negatif, atau tidak ada korelasi).

```
>x=normal(1,150); plot2d(x,x+rotright(x),>points,style="."):
```

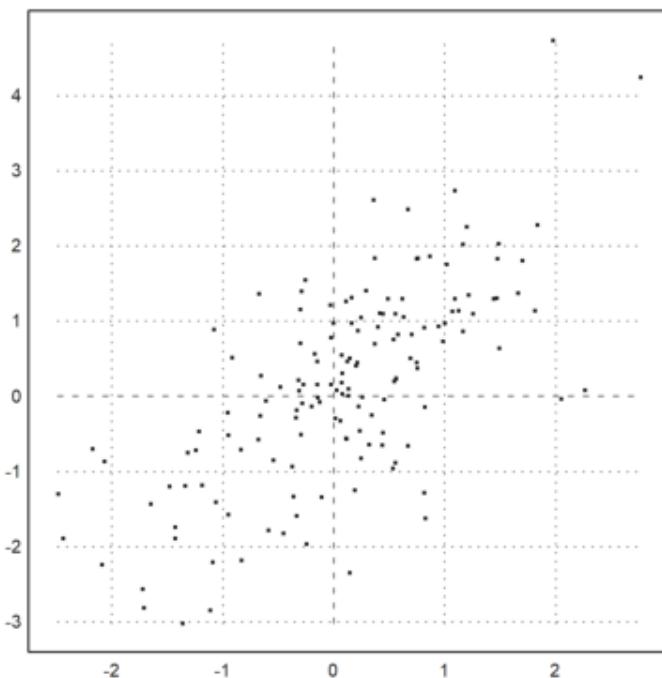


Diagram titik di atas memvisualisasikan data yang dihasilkan oleh fungsi normal dengan parameter mean 1 dan standar deviasi 150. Diagram di atas juga menunjukkan adanya korelasi yang positif karena pergeseran data tersebut ke kanan.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- $x=\text{normal}(1,150)$  menghasilkan data dari distribusi normal dengan mean 1 dan standar deviasi 150.
- $\text{plot2d}(x,x+\text{rotright}(x),>\text{points},\text{style}=".")$  digunakan untuk membuat plot 2D dari data  $x$  dan  $x+\text{rotright}(x)$  sebagai sumbu x dan y.  $>\text{points}$  digunakan untuk menunjukkan bahwa plot yang dihasilkan berupa titik-titik, dan  $\text{style}=". "$  menentukan gaya dari titik-titik tersebut.

```
>plot2d(normal(1500),normal(1500),>points,grid=6,style="."):
```

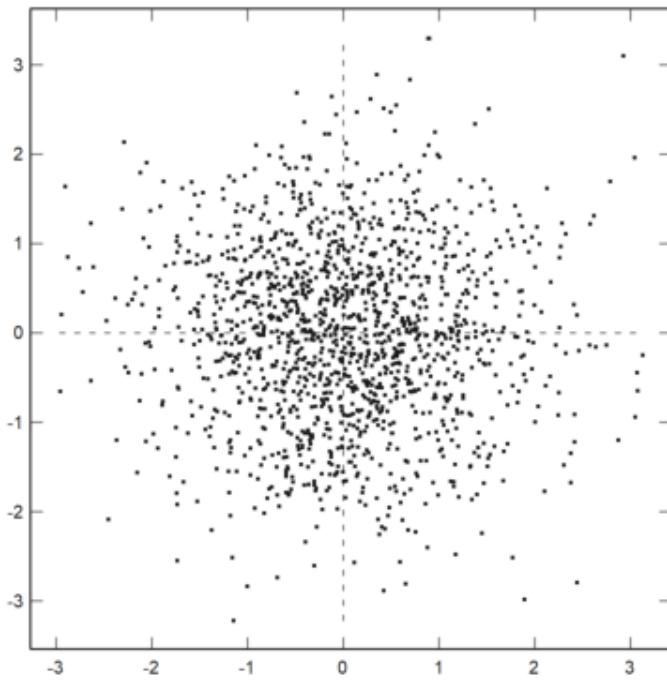


Diagram titik di atas memvisualisasikan distribusi dari dua set data yang dihasilkan oleh distribusi normal.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

`plot2d(normal(1500),normal(1500),>points,grid=6,style="..")` digunakan untuk membuat plot 2D dari dua distribusi normal yang menghasilkan masing-masing 1500 data. `>points` menunjukkan bahwa plot yang dihasilkan berupa titik-titik, `grid=6` menentukan ukuran grid, dan `style=".."` menentukan gaya dari titik-titik tersebut.

```
>{MS,hd}:=readtable("table1.dat",tok2:=[ "m", "f" ]); ...
>writetable(MS,labc=hd,tok2:=[ "m", "f" ]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

Tabel di atas memvisualisasikan data yang berisi mengenai survei anak, jenis kelamin mereka, usia mereka, usia orang tua mereka dan jumlah saudara kandung mereka.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- `{MS,hd}:=readtable("table1.dat",tok2:=[ "m", "f"])` digunakan untuk membaca data dari file "table1.dat" dan menyimpannya dalam variabel MS dan hd. `tok2:=[ "m", "f"]` menunjukkan bahwa data dalam file tersebut memiliki format "m" dan "f".

- `writetable(MS,labc=hd,tok2:=[ "m", "f"])` digunakan untuk menulis kembali data ke file dengan menggunakan label hd sebagai nama kolom dan format data "m" dan "f".

```
>scatterplots(tablecol(MS, 3:5),hd[3:5]):
```

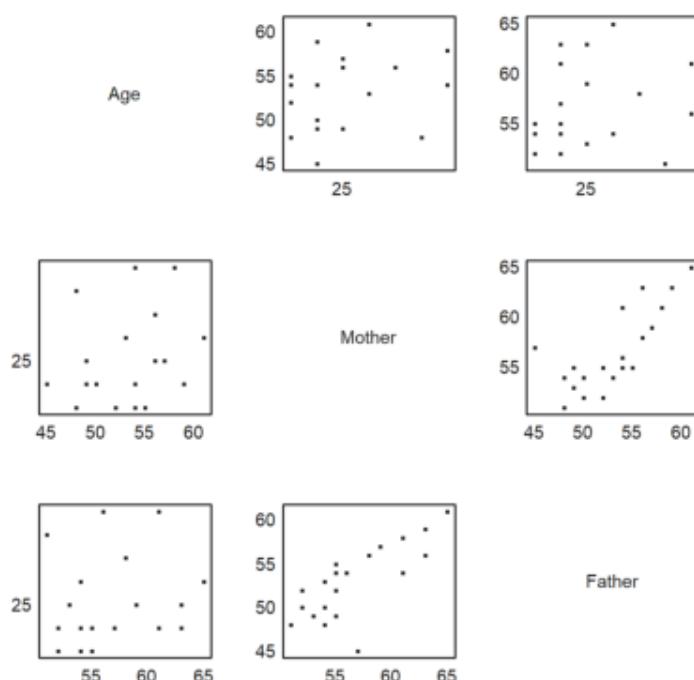


Diagram titik di atas memvisualisasikan hubungan antara tiga variabel dalam tabel data "MS", yaitu usia anak, ibu, dan ayah.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- `scatterplots(tablecol(MS,3:5),hd[3:5])` digunakan untuk membuat scatter plot (diagram titik) dari tiga kolom data dalam tabel MS dengan menggunakan label hd sebagai nama kolom.

- `tablecol(MS,3:5)` digunakan untuk memilih tiga kolom data dari tabel MS.

- `hd[3:5]` digunakan untuk memilih tiga label kolom dari hd.

---

## Contoh Soal

---

Tabel berikut ini memberikan informasi mengenai kandungan gula (gram) dan jumlah kalori dalam satu sajian dari 13 sampel merekereal.

```

>DATA:=[ ...
>1,4,120; ...
>2,15,200; ...
>3,12,140; ...
>4,11,110; ...
>5,8,120; ...
>6,6,80; ...
>7,14,170; ...
>8,2,100; ...
>9,7,130; ...
>10,14,190; ...
>11,20,190; ...
>12,3,110; ...
>13,13,120];
>VAR:= ["MERK", "GULA(gram)", "KALORI"];
>writetable(DATA, labc=VAR);

```

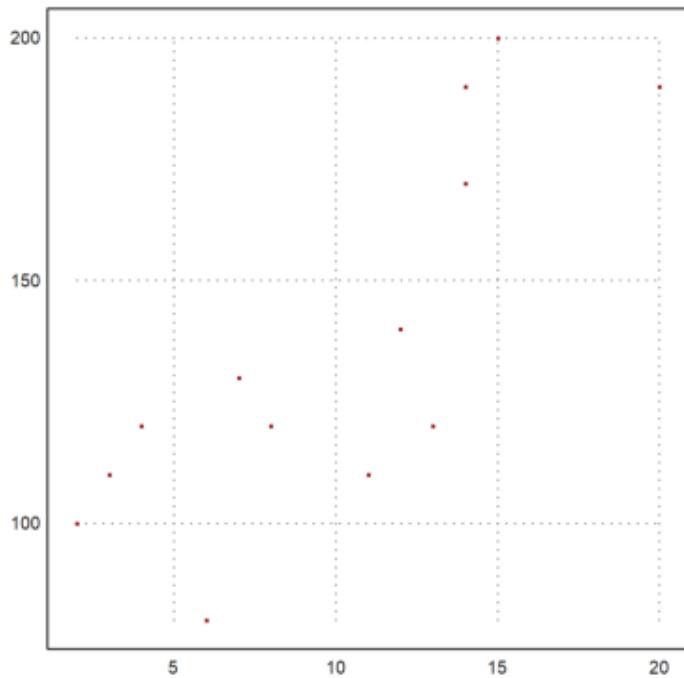
MERK	GULA (gram)	KALORI
1	4	120
2	15	200
3	12	140
4	11	110
5	8	120
6	6	80
7	14	170
8	2	100
9	7	130
10	14	190
11	20	190
12	3	110
13	13	120

- a. Gambarkan diagram titik atau scatter plot dari data di atas.  
b. Bagaimana pola penyebaran titik-titik yang telah digambar pada diagram di atas?  
c. Kesimpulan seperti apa yang dapat kalian ambil mengenai hubungan antara gula (gram) dan jumlah kalori?
- Penyelesaian:

```

>GULA:=[4,15,12,11,8,6,14,2,7,14,20,3,13];
>KALORI:=[120,200,140,110,120,80,170,100,130,190,190,110,120];
>plot2d(GULA,KALORI,>points,color=red,style="..");

```



- a. Diagram titik atau scatter plot di atas merupakan hasil visualisasi dari tabel data yang telah disebutkan sebelumnya.
  - b. Pada diagram di atas dapat terlihat bahwa pola penyebaran titiknya mempunyai kecenderungan semakin naik ke atas jika dilihat dari kiri bawah ke kanan atas.
  - c. Kesimpulan yang dapat diambil yaitu semakin tinggi kandungan gula maka semakin tinggi jumlah kalorinya.
- 

Diagram Garis Diagram garis atau bisa disebut line chart adalah jenis grafik statistik yang digunakan untuk menunjukkan perubahan atau tren sepanjang waktu atau variabel independen lainnya. Ini adalah salah satu cara yang paling umum digunakan untuk memvisualisasikan data berurutan dalam statistika.

Dalam diagram garis, data diplotkan sebagai titik-titik dan kemudian dihubungkan dengan garis lurus. Ini membantu untuk melihat perubahan atau tren dari satu titik ke titik berikutnya dengan jelas. Diagram garis sangat berguna untuk menyoroti pola, tren, atau fluktuasi dalam data sepanjang periode tertentu.

Contoh umum penggunaan diagram garis melibatkan waktu di sumbu horizontal (x) dan nilai atau frekuensi di sumbu vertikal (y). Misalnya, diagram garis dapat digunakan untuk menunjukkan perubahan suhu sepanjang waktu, pertumbuhan populasi, atau kinerja keuangan perusahaan sepanjang beberapa kuartal.

Untuk memetakan data, kita mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam jumlah kursi.

```
>BW := [...  
>1990, 662, 319, 239, 79, 8, 17; ...  
>1994, 672, 294, 252, 47, 49, 30; ...  
>1998, 669, 245, 298, 43, 47, 36; ...  
>2002, 603, 248, 251, 47, 55, 2; ...  
>2005, 614, 226, 222, 61, 51, 54; ...  
>2009, 622, 239, 146, 93, 68, 76; ...  
>2013, 631, 311, 193, 0, 63, 64];
```

Sintaks tersebut digunakan untuk membuat matriks BW yang berisi data-data tahun dan jumlah kursi untuk beberapa partai politik pada setiap tahun.

```
>P := ["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Sintaks tersebut digunakan untuk membuat array P yang berisi label-label untuk setiap kolom pada matriks BW.

```
>BT := BW[, 3:7]; BT := BT / sum(BT); YT := BW[, 1]';
```

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- BT:=BW[,3:7]; BT:=BT/sum(BT) digunakan untuk mengambil kolom ke-3 hingga ke-7 dari matriks BW, kemudian membagi setiap nilai dalam matriks tersebut dengan jumlah total nilai dalam matriks tersebut. Kolom 3 sampai 7 adalah jumlah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi seluruhnya. Sedangkan kolom 1 adalah tahun pemilihan.
- YT:=BW[,1]' digunakan untuk mengambil transpose dari kolom pertama matriks BW dan menyimpannya dalam variabel YT.

```
>writetable(BT * 100, wc=6, dc=0, >fixed, labc=P, labr=YT)
```

	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10

Sintaks tersebut digunakan untuk menulis matriks BT yang telah dikalikan dengan 100 ke dalam sebuah file. Parameter wc=6 menunjukkan lebar atau sekat kolom, dc=0 menunjukkan jumlah digit di belakang koma, >fixed menunjukkan bahwa jumlah digit di belakang koma tetap, labc=P menunjukkan label untuk kolom, dan labr=YT menunjukkan label untuk baris.

```
>BT1 := (BT[, 1; 1; 0; 0; 0])' * 100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Sintaks di atas digunakan untuk mengalikan baris pertama dari matriks BT dengan 100 dan menyimpan hasilnya dalam variabel BT1. Angka-angka dalam tanda kurung siku menunjukkan indeks baris yang ingin diambil dari matriks BT (pada sintaks tersebut indeks baris yang ingin diambil yaitu baris pertama), sedangkan tanda kutip di sebelah kanan menunjukkan bahwa kita ingin mengalikan baris tersebut dengan 100.

```
>statplot(YT, BT1, "b") :
```

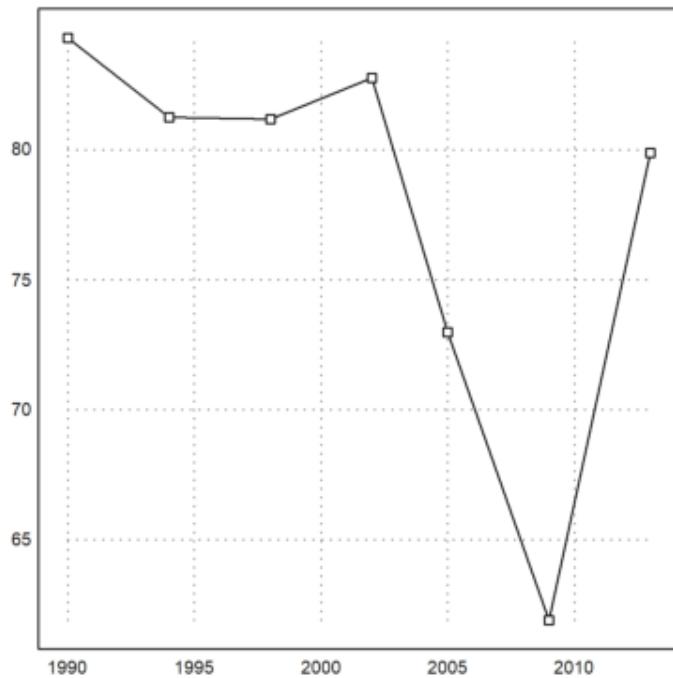


Diagram di atas memvisualisasikan persentase perolehan jumlah kursi setiap partai di setiap tahunnya, yaitu dari tahun 1990-2013.

Tipe plotplot yang tersedia adalah sebagai berikut:

- 'p' : plot titik
- 'l' : plot garis
- 'b' : keduanya (titik dan garis)
- 'h' : alur histogram
- 's' : plot permukaan

### Contoh Soal

Perhatikan tabel di bawah ini!

```
>BARANG:=[ ...
>2015,13000,10000,2700,10700; ...
>2020,15000,12500,4000,12300];
>VRB := ["TAHUN","SUSU(kaleng)","GULA(kg)", "JAGUNG(kg)", "BERAS(kg)"];
>writetable(BARANG,labc=VRB);
```

TAHUN	SUSU(kaleng)	GULA(kg)	JAGUNG(kg)	BERAS(kg)
2015	13000	10000	2700	10700
2020	15000	12500	4000	12300

Hitunglah persentase kenaikan harga dari tahun 2015 hingga tahun 2020 (jika penghitungan indeks harga menggunakan metode aggregatif sedehana) dan gambarkan grafik kenaikan harganya menggunakan diagram garis!

Penyelesaian:

```
>HARGA:=BARANG[,2:5]; TAHUN:=BARANG[,1]';  
>Po:=(13000+10000+2700+10700) // total harga barang di tahun 2015
```

36400

```
>Pn:=(15000+12500+4000+12300) // total harga barang di tahun 2020
```

43800

```
>Persentase:=(Pn/Po)*100
```

120.32967033

```
>Kenaikan:=(120.3-100)
```

20.3

Berdasarkan penghitungan tersebut, dapat disimpulkan bahwa dari tahun 2015 ke tahun 2020 terdapat kenaikan harga sebesar 20,3%.

```
>P1:=[36400,43800];  
>statplot(TAHUN,P1,"b"):
```

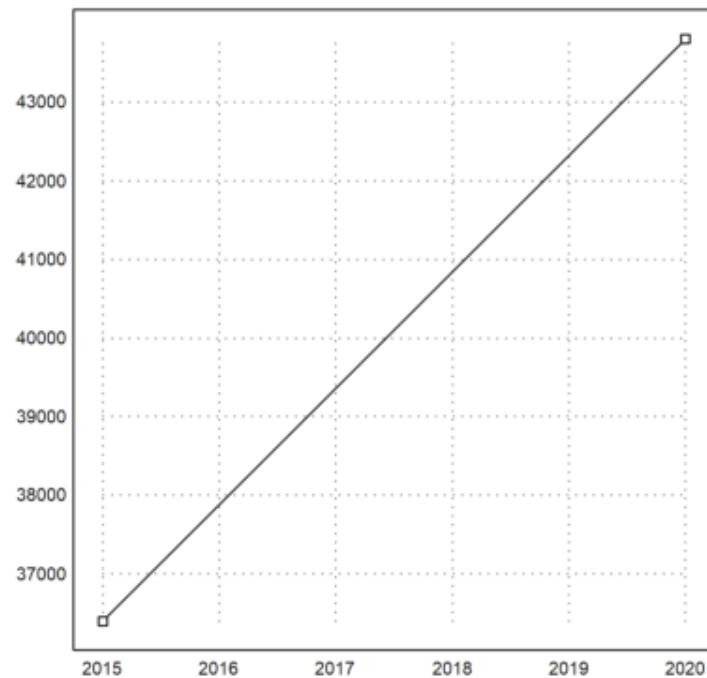


Diagram garis di atas memvisualisasikan kenaikan harga barang dari tahun 2015 hingga tahun 2020.

---

Kurva Regresi Kurva regresi adalah representasi grafis dari model regresi yang digunakan untuk memodelkan hubungan antara variabel independen (X) dan variabel dependen (Y). Persamaan regresi dapat digunakan untuk memprediksi atau mengestimasi nilai dari variabel tak bebas berdasarkan informasi dari variabel bebas. Persamaan regresi linear merupakan suatu persamaan yang berupa garis lurus, sedangkan persamaan regresi nonlinear bukan merupakan persamaan garis lurus.

Persamaan regresi linier umumnya ditulis sebagai  $Y=a+bX$ , di mana:

$Y$  = garis regresi/ variable response

$a$  = konstanta (intercept), perpotongan dengan sumbu-y

$b$  = konstanta regresi (slope), kemiringan

$X$  = variabel bebas/ predictor

Regresi linier dapat dilakukan dengan fungsi `polyfit()` atau berbagai fungsi fit. Sebagai permulaan, kami mencari garis regresi untuk data univariat dengan `polyfit(x,y,1)`.

```
>x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x' | y',labc=["x","y"])
```

x	y
1	2
2	3
3	1
4	5
5	6
6	3
7	7
8	8
9	9
10	8

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- `x=1:10; y=[2,3,1,5,6,3,7,8,9,8];`: Perintah ini digunakan untuk membuat vektor x yang berisi bilangan bulat dari 1 hingga 10, dan vektor y yang berisi nilai-nilai acak.

- `writetable(x' | y',labc=["x","y"]);`: Perintah ini digunakan untuk membuat tabel dua kolom dari dua vektor x dan y. Parameter `x' | y'` menunjukkan bahwa vektor x dan y akan disusun secara vertikal, sedangkan `labc=["x","y"]` mengatur label kolom tabel.

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. Pertama koefisien kecocokan linier.

```
>p=polyfit(x,y,1)
```

```
[ 0.733333, 0.812121]
```

Sintaks di atas digunakan untuk melakukan fitting kurva polinomial orde satu pada data yang diberikan. Parameter x dan y adalah data yang akan di-fit, sedangkan 1 menunjukkan orde polinomial yang digunakan. Hasil fitting akan disimpan dalam variabel p.

Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[ 4.71566, 0.38319]
```

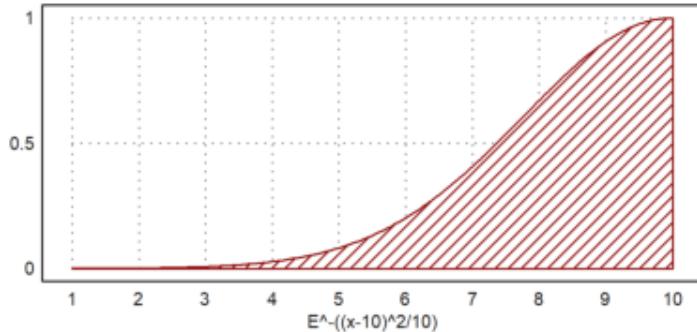
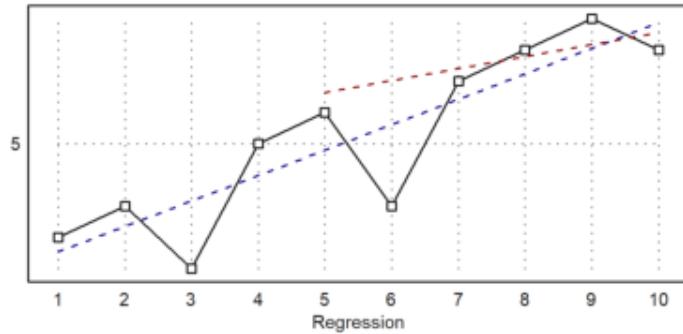
Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- `w &= "exp(-(x-10)^2/10)":` digunakan untuk mendefinisikan fungsi berat (weight function) w sebagai  $\exp(-(x-10)^2/10)$ . Fungsi berat digunakan dalam fitting polinomial untuk memberikan bobot pada setiap titik data, sehingga titik-titik data yang lebih dekat dengan nilai 10 akan diberi bobot yang lebih tinggi.

- `pw=polyfit(x,y,1,w=w(x)):` digunakan untuk melakukan fitting polinomial pada data yang diberikan dengan menggunakan fungsi berat w. Fungsi polyfit digunakan untuk menemukan koefisien polinomial yang sesuai dengan data. Dalam kasus ini, fitting dilakukan dengan polinomial derajat 1, dengan menggunakan fungsi berat yang telah didefinisikan sebelumnya.

Kami memasukkan semuanya ke dalam satu plot untuk titik dan garis regresi, dan untuk bobot yang digunakan.

```
>figure(2,1); ...
>figure(1); statplot(x,y,"b",xl="Regression"); ...
> plot2d("evalpoly(x,p)",>add,color=blue,style="--"); ...
> plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"); ...
>figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red, xl=w); ...
>figure(0):
```



Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- `figure(2,1);` digunakan untuk mengatur tampilan grafik dengan berbeda nomor tampilan (handle).

- `statplot(x,y,"b",xl="Regression"):` digunakan untuk menampilkan data (x,y) dengan label nama sumbu-x nya yaitu "Regression".

- `plot2d("evalpoly(x,p)",>add,color=blue,style="--"):` fungsi evalpoly digunakan untuk menghitung nilai polinomial pada titik-titik data yang diberikan. Dalam hal ini, evalpoly(x,p) menghitung nilai polinomial pada titik-titik data yang diberikan dari polinomial pertama (p). Hasilnya kemudian digunakan untuk menambahkan plot polinomial pertama ke grafik dengan warna biru dan gaya putus-putus.

- `plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"):` fungsi evalpoly digunakan untuk menghitung nilai polinomial yang diperoleh dari polyfit dengan fungsi berat. Dalam hal ini, evalpoly(x,pw) menghitung nilai polinomial dengan fungsi berat pada titik-titik data dari 5 hingga 10. Hasilnya kemudian digunakan untuk menambahkan plot polinomial dengan fungsi berat ke grafik dengan warna merah dan gaya putus-putus.

- `plot2d(w,1,10,>filled,style="/",fillcolor=red, xl=w):` sintaks tersebut digunakan untuk membuat plot dari area

di bawah kurva polinomial dengan fungsi berat w pada rentang 1 hingga 10, dengan gaya pengisian warna merah dan label sumbu x yang diberi nilai w

---

Menampilkan Data Frame Data frame biasanya merujuk pada struktur data tabular dua dimensi yang digunakan untuk menyimpan dan mengorganisir data. Data frame adalah konsep yang umumnya terkait dengan pemrograman statistik, terutama dalam bahasa seperti R dan Python (menggunakan pustaka pandas).

Di direktori buku catatan ini kita akan menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file tersebut. Datanya berasal dari buku online Jerman "Einführung in die Statistik mit R" yang dibuat oleh A. Handl.

```
>printfile("table.dat", 4);  
  
Person Sex Age Titanic Evaluation Tip Problem  
1 m 30 n . 1.80 n  
2 f 23 y g 1.80 n  
3 f 26 y g 1.80 y
```

Sintaks tersebut digunakan untuk mencetak isi file "table.dat" pada empat baris pertama.

Tabel berisi 7 kolom angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan terjemahan kami sendiri untuk token.

Untuk ini, kami mendefinisikan set token. Fungsi strtokens() mendapatkan vektor string token dari string yang diberikan.

```
>mf:=[ "m" , "f" ] ; yn:=[ "y" , "n" ] ; ev:=strtokens( "g vg m b vb" );
```

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- mf:=[ "m" , "f" ]: Mendefinisikan variabel mf sebagai array yang berisi dua elemen, yaitu "m" dan "f". Kemungkinan ini digunakan untuk merepresentasikan jenis kelamin, di mana "m" mewakili laki-laki dan "f" mewakili perempuan.
- yn:=[ "y" , "n" ]: Mendefinisikan variabel yn sebagai array yang berisi dua elemen, yaitu "y" dan "n". Kemungkinan ini digunakan untuk merepresentasikan jawaban ya ("y") dan tidak ("n") dari suatu pertanyaan.
- ev:=strtokens( "g vg m b vb" ): Mendefinisikan variabel ev sebagai hasil dari pemisahan string "g vg m b vb" berdasarkan spasi. Kemungkinan ini digunakan untuk merepresentasikan kategori tertentu yang terkait dengan data atau survei.

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen tok2, tok4 dll. adalah terjemahan dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda harus menyediakannya dengan ":".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn) ;
```

Sintaks ini digunakan untuk membaca data dari file "table.dat" ke dalam sebuah tabel. Pada saat membaca file, argumen opsional tok2, tok4, tok5, dan tok7 digunakan untuk menentukan bagaimana data dalam file tersebut akan diinterpretasikan.

```
>load over statistics;
```

Sintaks ini digunakan untuk memuat data atau variabel yang telah disimpan sebelumnya. Untuk mencetak, kita perlu menentukan set token yang sama. Kami mencetak empat baris pertama saja.

```
>writetable(MT[1:4],labc=hd,wc=6,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan terlebih dahulu, kita hanya perlu menentukan, kolom mana yang berisi token dan bukan angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

Fungsi readtable() sekarang mengembalikan satu set token.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:

- ctok=[2,4,5,7]: digunakan untuk mendefinisikan variabel ctok sebagai array yang berisi angka 2, 4, 5, dan 7. Kemungkinan ini digunakan untuk menentukan kolom-kolom tertentu yang akan dibaca dari file "table.dat".
- {MT,hd,tok}=readtable("table.dat",ctok=ctok): sintaks ini menggunakan fungsi readtable untuk membaca data dari file "table.dat" ke dalam tiga variabel yang berbeda, yaitu MT, hd, dan tok, dengan menggunakan kolom-kolom yang telah ditentukan sebelumnya dalam variabel ctok.

```
>tok
```

```
m  
n  
f  
y  
g  
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke angka.

String khusus NA="." ditafsirkan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAvl.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Sintaks tersebut digunakan untuk mengakses elemen pertama dari variabel MT yang telah diimpor sebelumnya.

Berikut isi tabel dengan nomor yang belum diterjemahkan.

```
>writetable(MT,wc=5)
```

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4
7	3	31	4	6	2.8	2
8	1	23	2	.	0.8	2
9	3	24	4	6	1.8	4
10	1	26	2	.	1.8	2
11	3	23	4	6	1.8	4
12	1	32	4	5	1.8	2
13	1	29	4	6	1.8	4
14	3	25	4	5	1.8	4
15	3	31	4	5	0.8	2
16	1	26	4	5	2.8	2
17	1	37	2	.	3.8	2
18	1	38	4	5	.	2
19	3	29	2	.	3.8	2
20	3	28	4	6	1.8	2
21	3	28	4	1	2.8	4
22	3	28	4	6	1.8	4
23	3	38	4	5	2.8	2
24	3	27	4	1	1.8	4
25	1	27	2	.	2.8	4

Untuk kenyamanan, Anda dapat memasukkan keluaran readtable() ke dalam list.

```
>Table={{readtable("table.dat",ctok=ctok)}},
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll atau menggunakan tabel daftar

```
>writetable(Table,ctok=ctok,wc=5);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n

18	m	38	y	g	.	n
19	f	29	n	.	3.8	n
20	f	28	y	vg	1.8	n
21	f	28	y	m	2.8	y
22	f	28	y	vg	1.8	y
23	f	38	y	g	2.8	n
24	f	27	y	m	1.8	y
25	m	27	n	.	2.8	y

Fungsi `tablecol()` mengembalikan nilai kolom tabel, melewatan baris apa pun dengan nilai NAN ("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Sintaks ini digunakan untuk mengekstrak kolom ke-5 dan ke-6 dari matriks atau tabel MT. Hasilnya akan disimpan dalam variabel c dan i secara berturut-turut. Dengan kata lain, c akan berisi kolom ke-5 dari MT, sedangkan i akan berisi kolom ke-6 dari MT.

Kita bisa menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

Person	Evaluation	Tip
2	g	1.8
3	g	1.8
6	g	2.8
7	vg	2.8
9	vg	1.8
11	vg	1.8
12	g	1.8
13	vg	1.8
14	g	1.8
15	g	0.8
16	g	2.8
20	vg	1.8
21	m	2.8
22	vg	1.8
23	g	2.8
24	m	1.8

Berikut adalah maksud dari sintaks tersebut:

- `MT[i,j]`: Mengambil subset dari tabel MT yang terdiri dari baris-baris yang ditentukan oleh i dan kolom-kolom yang ditentukan oleh j. Ini mungkin digunakan untuk membuat subset dari data yang akan ditulis ke dalam file.

- `labc=hd[j]`: Menentukan label kolom untuk subset kolom yang dipilih dari MT berdasarkan label kolom dari `hd`. Ini mungkin digunakan untuk menetapkan label yang sesuai untuk kolom-kolom yang akan ditulis ke dalam file.

- `ctok=[2]`: Menentukan token tertentu yang akan digunakan saat menulis tabel ke dalam file. Ini mungkin digunakan untuk menetapkan cara khusus untuk menginterpretasikan data saat ditulis ke dalam file.

- `tok=tok`: Menggunakan token yang telah ditentukan sebelumnya saat menulis tabel ke dalam file. Ini mungkin digunakan untuk memastikan bahwa token yang sama digunakan saat menulis dan membaca tabel.

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Tabel dalam kasus ini.

```
>MT=Table[1];
```

Tentu saja, kita juga dapat menggunakannya untuk menentukan nilai rata-rata suatu kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT, 6))
```

2.175

Fungsi getstatistics() mengembalikan elemen dalam vektor, dan jumlahnya.

```
>{xu, count}=getstatistics(tablecol(MT, 5)); xu, count,
```

```
[1, 5, 6]
[2, 9, 6]
```

Variabel xu adalah elemen unik (indeks) dari kolom ke-5 tabel MT, sedangkan variabel count adalah jumlah data dari variabel xu.

Kita bisa mencetak hasilnya di tabel baru.

```
>writetable(count', labr=tok[xu])
```

m	2
g	9
vg	6

Fungsi selecttable() mengembalikan tabel baru dengan nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama kita mencari indeks dari dua nilai kita di tabel token.

```
>v:=indexof(tok, ["g", "vg"])
```

```
[5, 6]
```

Sekarang kita dapat memilih baris tabel, yang memiliki salah satu nilai v pada baris ke-5.

```
>MT1:=MT[selectrows(MT, 5, v)]; i:=sortedrows(MT1, 5);
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstraksi dan diurutkan di kolom ke-5.

```
>writetable(MT1[i], labc=hd, ctok=ctok, tok=tok, wc=7);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
6	m	28	y	g	2.8	y
18	m	38	y	g	.	n

16	m	26	y	g	2.8	n
15	f	31	y	g	0.8	n
12	m	32	y	g	1.8	n
23	f	38	y	g	2.8	n
14	f	25	y	g	1.8	y
9	f	24	y	vg	1.8	y
7	f	31	y	vg	2.8	n
20	f	28	y	vg	1.8	n
22	f	28	y	vg	1.8	y
13	m	29	y	vg	1.8	y
11	f	23	y	vg	1.8	y

Untuk statistik selanjutnya, kami ingin menghubungkan dua kolom tabel. Jadi kita ekstrak kolom 2 dan 4 dan urutkan tabelnya.

```
>i=sortedrows(MT,[2,4]); ...
>writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

m	n
m	n
m	n
m	n
m	n
m	n
m	n
m	n
m	y
m	y
m	y
m	y
m	y
f	n
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y

Dengan `getstatistics()`, kita juga bisa menghubungkan jumlah dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...
>{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
>writetable(count,labr=tok[xu1],labc=tok[xu2])
```

	n	y
m	7	5
f	1	12

Sebuah tabel dapat ditulis ke file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Kemudian kita bisa membaca tabel dari file tersebut.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
>writetable(MT2,labr=hdr,labc=hd)
```

	n	Y
m	7	5
f	1	12

Dan hapus file tersebut.

```
>fileremove(filename);
```

## Sub Topik 8 : Melakukan Perhitungan Untuk Uji Statistika

Perhitungan untuk Uji Statistika Dalam Euler, banyak uji yang diterapkan. Semua uji dalam Euler mengembalikan kesalahan yang diterima jika hipotesis nol ditolak.

Setiap jenis uji statistik memiliki asumsi dan syarat tertentu yang harus dipenuhi sebelum dapat diterapkan pada data. Pemilihan jenis uji statistik yang tepat sangat penting untuk memastikan hasil analisis data yang akurat dan dapat diandalkan.

### Simulasi Monte Carlo

Simulasi Monte Carlo yang digunakan untuk memecahkan masalah dengan menggunakan sampel acak berulang untuk memperkirakan distribusi dari suatu fenomena. Dalam konteks uji statistika, Monte Carlo dapat digunakan untuk menghitung nilai-nilai yang sulit atau tidak mungkin dihitung secara analitik, seperti p-value dalam uji chi-square.

Euler dapat digunakan untuk mensimulasikan kejadian acak. Berikut ini contoh yang mensimulasikan 1000 kali 3 lemparan dadu, dan ditampilkan distribusi jumlahnya.

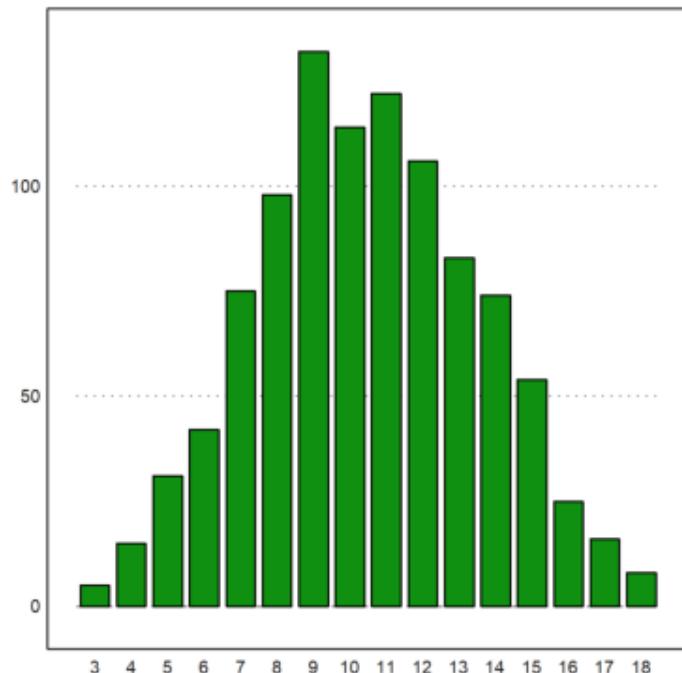
Fungsi ini akan menghasilkan 1000 bilangan bulat acak antara 3 dan 6, menjumlahkannya, dan menyimpan hasilnya dalam ds.

Fungsi getmultiplicities(v,x) mengembalikan kelipatan dari nilai-nilai dalam v dalam vektor x. Dalam hal ini, fs akan berisi kelipatan dari nilai 3 hingga 18 dalam vektor ds.

```
>ds:=sum(intrandom(1000,3,6))'; fs=getmultiplicities(3:18,ds)
```

```
[5, 15, 31, 42, 75, 98, 132, 114, 122, 106, 83, 74, 54,
25, 16, 8]
```

```
>columnsplot(fs,lab=3:18):
```



Untuk menentukan distribusi yang diharapkan tidak begitu mudah. Diperlukan rekursi lanjutan untuk hal ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat dinyatakan sebagai jumlah dari n bilangan dalam rentang 1 sampai m. Hal berikut bekerja secara rekursif dengan cara yang jelas.

```
>function map countways (k; n, m) ...
```

```

if n==1 then return k>=1 && k<=m
else
  sum=0;
  loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
  return sum;
end;
endfunction
```

```
>cw=countways(3:18,3,6)
```

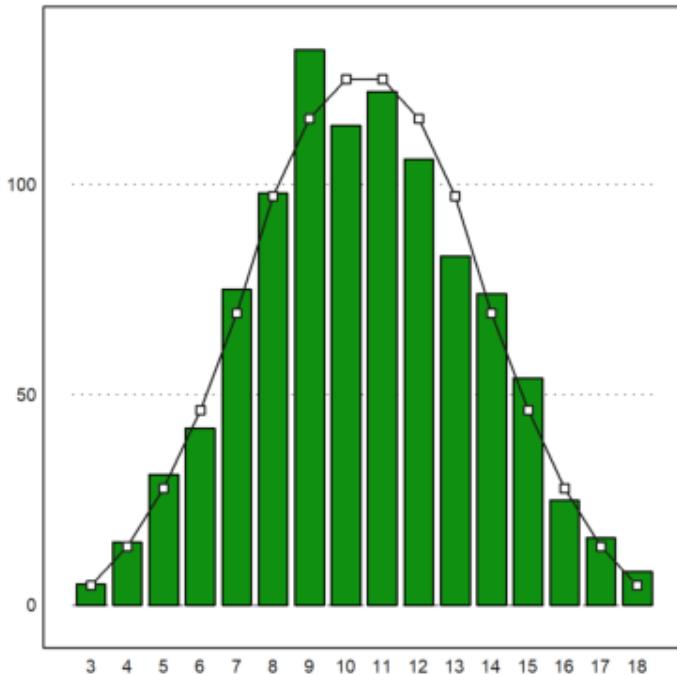
```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,
1]
```

Nilai yang diharapkan dapat direpresentasikan ke dalam plot.

Fungsi countways menghitung jumlah cara di mana bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 hingga m.

Fungsi ini bekerja secara rekursif dengan cara yang jelas. Jika n=1, maka fungsi mengembalikan  $k \geq 1 \ \&\& k \leq m$ , jika tidak, fungsi akan menghitung jumlah cara dengan melakukan iterasi dari 1 hingga m. Hasilnya adalah jumlah cara di mana bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 hingga m.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



## Uji Chi Square

Untuk melakukan uji chi<sup>2</sup> (chi-squared) kita menggunakan fungsi chitest(). Fungsi ini digunakan untuk membandingkan frekuensi observasi dengan frekuensi yang diharapkan. Dalam konteks ini, vektor x merepresentasikan frekuensi observasi, sedangkan vektor y merepresentasikan frekuensi yang diharapkan. Misalnya, jika dari sampel 100 orang ditemukan 40 pria, maka vektor observasi x adalah [40,60], dan vektor harapan y mungkin [50,50]. Uji chi<sup>2</sup> digunakan untuk menilai sejauh mana sampel tersebut sesuai dengan frekuensi yang diharapkan.

Sebagai contoh, kami menguji lemparan dadu untuk distribusi baju. Pada 600 lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6))
```

0.498830517952

Fungsi ini digunakan untuk menguji asosiasi antara dua set data. Set pertama [90,103,114,101,103,89] mewakili jumlah yang diamati, dan set kedua dup(100,6) mewakili jumlah yang diharapkan. Hasil dari fungsi chitest adalah probabilitas yang terkait dengan statistik uji chi-kuadrat, yang menunjukkan kemungkinan bahwa data kategoris yang diamati diambil dari distribusi yang diharapkan

Tes chi-kuadrat juga memiliki mode, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harus hampir sama. Parameter >p menginterpretasikan vektor-y sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.497

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak distribusi baju. Ini tidak membuktikan bahwa dadu kami adil. Tapi kita tidak bisa menolak hipotesis kita.

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan tes yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.594554930686

Fungsi random([1,n\*6]) digunakan untuk menghasilkan serangkaian bilangan acak antara 1 dan 6 sebanyak n kali. Kemudian, fungsi count(t\*6,6) digunakan untuk menghitung berapa kali angka 6 muncul dalam serangkaian bilangan acak tersebut. Fungsi dup(n,6) digunakan untuk menghasilkan serangkaian bilangan 6 sebanyak n kali. Hasil dari fungsi chitest(count(t\*6,6),dup(n,6)) akan mengembalikan nilai p, yang dapat digunakan untuk menentukan apakah serangkaian bilangan acak tersebut terdistribusi secara merata atau tidak.

## Uji T

---

Uji-t adalah sebuah metode statistik yang digunakan untuk menguji perbedaan signifikan antara rata-rata dua kelompok yang berbeda.

Misalnya, akan dilakukan pengujian nilai rata-rata dari 100 elemen data menggunakan uji-t. Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

0.161169176307

Fungsi ttest() membutuhkan nilai rata-rata, simpangan, jumlah data, dan nilai rata-rata yang akan diuji. Dalam syntax tersebut, normal([1,100]) menghasilkan 100 angka acak yang diambil dari distribusi normal dengan nilai rata-rata 0 dan simpangan baku 1. Kemudian, nilai-nilai tersebut dikalikan dengan 10 dan ditambahkan dengan 200 untuk menghasilkan sampel data dengan nilai rata-rata 200 dan simpangan baku 10.

Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kami menolak hipotesis bahwa mereka memiliki rata-rata yang sama, jika hasilnya <0,05.

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.23564235612

Jika kita menambahkan bias ke satu distribusi, kita mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

0.000114481122387

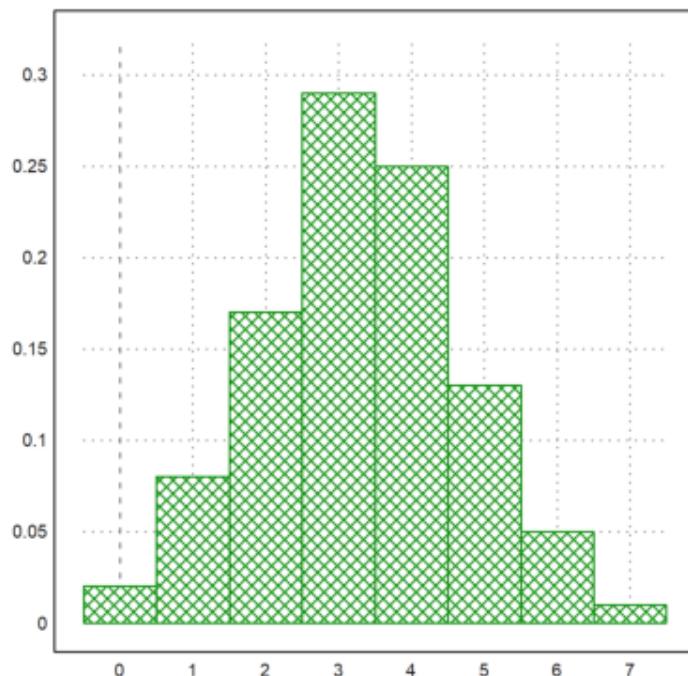
Pada contoh berikutnya, kita menghasilkan 20 lemparan dadu acak sebanyak 100 kali dan menghitung yang ada di dalamnya. Harus ada  $20/6=3,3$  yang rata-rata.

```
>R=random(100,20); R=sum(R*6<=1); mean(R)
```

3.31

Bandingkan jumlah satu dengan distribusi binomial. Pertama lakukan plot distribusi satuan.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\\"/"):
```



```
>t=count(R,21);
```

Menghitung nilai yang diharapkan

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-kuadrat menolak hipotesis bahwa distribusi kami adalah distribusi binomial, jika hasilnya <0,05.

```
>chitest(t1,b1)
```

0.45755878393

## Uji Independensi

---

Uji independensi adalah salah satu uji statistik yang digunakan untuk mengetahui apakah ada hubungan antara dua variabel kategorik, dengan kata lain untuk mengetahui independensi antara variabel baris dan kolom. Uji ini berguna untuk mengukur perbedaan pengamatan dan menaksir frekuensi suatu pengamatan dalam kategori tertentu.

Contoh berikut berisi hasil dua kelompok orang (laki-laki dan perempuan, katakanlah) memberikan suara untuk 6 jenis minuman kaleng

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...  
> writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Tes tabel chi^2 dapat melakukan uji independensi suara dari jenis kelamin

```
>tabletest(A)
```

0.990701632326

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
>writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	24.9	37.9	41.9	50.3	63.3	74.7
f	25.1	38.1	42.1	50.7	63.7	75.3

Kita dapat menghitung koefisien kontingensi yang dikoreksi. Karena sangat dekat dengan 0, kami menyimpulkan bahwa pemilihan jenis minuman kaleng tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

0.0427225484717

## Metode Anova

---

Selanjutnya kami menggunakan analisis varians (Uji-F) untuk menguji tiga sampel data yang terdistribusi normal untuk nilai rata-rata yang sama. Metode tersebut disebut ANOVA (analisis varians). Di Euler, fungsi varanalysis() digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1)
```

106.545454545

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

119.111111111

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

116.3

```
>varanalysis(x1,x2,x3)
```

0.0138048221371

Dalam pengujian hipotesis statistik di atas, hipotesis nol mengasumsikan bahwa nilai rata-rata ke-3 set data ( $x_1, x_2, x_3$ ) sama. Namun, berdasarkan analisis varians telah kita lakukan, hipotesis nol dapat ditolak dengan probabilitas kesalahan 1,3%. Ini berarti bahwa ada perbedaan yang signifikan secara statistik antara rata-rata dari ketiga set angka.

## Sub Topik 9 : Menyimpan Data Hasil Analisis

Data-data yang kita gunakan dalam melakukan analisis statistika dapat kita simpan ke dalam suatu file sehingga ketika kelak ingin digunakan lagi, data tersebut masih ada di file penyimpanan kita. Tak hanya itu, hasil dari analisis statistika yang sudah kita lakukan pun dapat kita simpan ke dalam suatu file.

Berikut adalah cara menyimpan/menulis data ke suatu file.

```
>a=random(1,100); mean(a); dev(a);
>filename="Simpan";
```

Selelah memberi nama untuk file, kita akan menulis vektor a ke dalam file dengan menggunakan fungsi writematrix() dan menggunakan fungsi readmatrix() untuk membaca data.

```
>writematrix(a',filename);
>a=readmatrix(filename');
```

Kita juga bisa menghapus file yang sudah tersimpan dengan menggunakan fileremove

```
>fileremove(filename);
```

Kemudian kita akan mencoba untuk menggantikan data baru ke file lama dengan menghapus semua data lama, dan menulis lagi data baru yang akan disimpan.

```
>file="Simpan"; open(file, "w");
>writeln("A,B,C"); writematrix(random(3,3));
>close();
>printfile(file)
```

```
A,B,C
0.8351051327697636,0.08458248162153686,0.5192250558799737
0.8548977070796793,0.7679427770316303,0.4018472121828296
0.02466356619093713,0.4253015574769268,0.1649367711598173
```

Selain itu kita juga bisa menyimpan dalam bentuk excel

```
>file="test.csv";
>M=random(3,3); writematrix(M,file);
```

Berikut adalah isi dari file ini.

```
>printfile(file)
```

```
0.2641349455581374,0.6713124187949838,0.135806558906826
0.3437161954193733,0.7730785630232085,0.05730568239045469
0.3519819858104311,0.1107098010416762,0.4220892269525604
```

CSV ini dapat dibuka pada sistem bahasa Inggris ke dalam Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Tetapi titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari file dengan readmatrix().

```
>readmatrix(file)
```

```
0.264135      0.671312      0.135807
0.343716      0.773079      0.0573057
0.351982      0.11071       0.422089
```