Tugas Latex Aplikasi Komputer

Akhmad Rizki Khamid
22305141008
Matematika E 2022

PRODI MATEMATIKA
DEPARTEMEN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2023

DAFTAR ISI

BAB 1

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

[a4paper,10pt]article eumat

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)
```

Menjabarkan:

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

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

Baris Perintah

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasilnya. 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 dijalankan sesuai 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 jalur dihubungkan dengan "..." kedua jalur akan selalu dijalankan 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 menyampaikan perintah panjang ke dua baris atau lebih. Anda dapat menekan Ctrl+Return untuk membagi baris menjadi dua pada kursor saat ini, atau Ctlr+Back untuk menggabungkan posisi baris.

Untuk melipat semua multi-garis tekan Ctrl+L. Maka garis-garis berikutnya hanya akan terlihat, jika salah satunya mendapat fokus. Untuk melipat satu multi-baris, memulai baris pertama dengan "%+".

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

Garis yang dimulai dengan %% tidak akan terlihat sama sekali.

```
81
```

Euler mendukung loop di baris perintah, asalkan cocok ke dalam satu baris atau multi-baris. Tentu saja, pembatasan ini tidak berlaku dalam program. Untuk informasi lebih lanjut lihat pendahuluan berikut.

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

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

```
>x := 1.5; // comments go here before the ...
>repeat xnew:=(x+2/x)/2; until xnew~=x; ...
>   x := xnew; ...
>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 bagian komentar di atas perintah untuk membuka perintah.

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

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

```
2.5
```

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

Euler mengetahui 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 sqrt di Euler. Tentu saja, x^(1/2) juga dimungkinkan.
Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pendahuluan ini menggunakan bentuk yang terakhir. Spasi tidak penting. Tapi jarak antar perintah diharapkan.
Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan keluaran 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 mendapatkan 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 formulir 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 yang mengakhiri tanda kurung tutup. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.
Hasil perhitungan ini berupa bilangan floating point. Ini secara default dicetak dengan akurasi sekitar 12 digit. Di baris perintah berikut, kita juga mempelajari 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, harus berisi tanda kurung untuk memaksakan urutan eksekusi yang benar. Jika ragu, memasang braket adalah ide yang bagus. Perhatikan bahwa EMT menampilkan 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 pangkat untuk a positif atau bilangan bulat b (a**b juga
```

berfungsi)

```
N! operator faktorial
```

dan masih banyak lagi.
Berikut beberapa fungsi yang mungkin Anda perlukan. Masih banyak lagi.

```
sin,cos,tan,atan,asin,acos,rad,deg
log,exp,log10,sqrt,logbase
bin,logbin,logfac,mod,floor,ceil,round,abs,sign
conj,re,im,arg,conj,real,complex
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 (tanda kurung bulat), setiap kali ada keraguan tentang urutan eksekusi! Berikut ini tidak sama dengan (2^3)^4, yang merupakan default untuk 2^3^4 di EMT (beberapa sistem numerik melakukannya dengan cara lain).

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

```
2.41785163923e+24
4096
2.41785163923e+24
```

Bilangan Nyata

Tipe data primer pada 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)
```

```
1.0101010101010101010101010101010101010101010101010101*2^-2
```

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

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

Strings

Sebuah string di 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 dalam hal ini diubah menjadi string.

```
>"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^2.
```

Fungsi print juga mengubah angka menjadi string. Ini dapat memerlukan sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan optimalnya satuan.

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

```
Golden Ratio : 1.61803
```

Ada string khusus none yang tidak dicetak. Itu dikembalikan oleh beberapa fungsi, ketika hasilnya tidak menjadi masalah. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan return.)

```
>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 [tidak ada]. 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
```

```
 = 45°
```

I

Di komentar, entitas yang sama seperti alpha;, beta; dll. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk Lateks. (Detail lebih lanjut 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")
```

```
[196,  32,  105,  115,  32,  97,  32,  71,  101,  114,  109,  97,  110,
32,  108,  101,  116,  116,  101,  114]
```

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

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

```
Ü is a German letter
```

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
```

```
We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

```
>u"&#196;hnliches"
```

```
Ähnliches
```

Nilai Boolean

Nilai Boolean diwakili dengan 1=true atau 0=false di 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 "dan" dan "atau" hanya dapat digunakan dalam kondisi "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 nonzeros() untuk mengekstrak elemen tertentu dari vektor. Dalam contoh ini, kita menggunakan kondisi isprime(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 keluaran default EMT mencetak 12 digit. Untuk memastikan bahwa kami melihat defaultnya, kami mengatur ulang formatnya.

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
      3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
    0.33333
    3.14159
    0.84147
```

The default is format(12).
Standarnya adalah format(12).

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

```
0.333333333333
```

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

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

```
    0.66     0.2    0.89    0.28    0.53    0.31    0.44     0.3
    0.28    0.88    0.27     0.7    0.22    0.45    0.31    0.91
    0.19    0.46   0.095     0.6    0.43    0.73    0.47    0.32
```

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

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

```
  3.1416
```

Fungsi "format terpanjang" juga mengatur format skalar.

```
>longestformat; pi
```

```
  3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran terpenting.

```
  format terpendek, format pendek, format panjang, format terpanjang
  format(panjang,digit) format bagus(panjang)
  frakformat(panjang)
  merusak format
```

Akurasi internal EMT adalah sekitar 16 tempat desimal, yang merupakan standar IEEE. Nomor disimpan dalam format internal ini.
Namun format keluaran EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
  3.141592653589793
```

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

```
    3.14159
```

Standarnya adalah defformat().

```
>defformat; // default
```

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

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

```
        4.934802200544679
```

Ada juga operator singkat untuk mencetak hasil dalam format pecahan. Kami sudah menggunakannya 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 secara tepat. Kesalahannya bertambah sedikit, seperti yang Anda lihat pada perhitungan berikut.

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

```
 -1.110223024625157e-16
```

Tetapi dengan "format panjang" default Anda tidak akan menyadarinya. Untuk kenyamanan, keluaran angka yang sangat kecil adalah 0.

```
>0.1+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 dengan EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda ingin menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "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, meskipun 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" selain 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
```

Sebagai referensi, kami mencatat 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)
```

```
45
```

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

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

```
12
```

Berdasarkan 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
```

Bentuk ekspresi khusus memungkinkan variabel apa pun sebagai parameter yang tidak disebutkan namanya untuk mengevaluasi ekspresi, bukan hanya "x", "y", dll. Untuk ini, mulailah ekspresi dengan "@(variabel) ...".

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

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

Hal ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang memerlukan ekspresi dalam "x".
Cara paling dasar untuk mendefinisikan suatu fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utamanya adalah x, ekspresi dapat dievaluasi seperti halnya fungsi.
Seperti yang Anda lihat pada 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 bersifat simbolis. Hal 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 memperhatikan bahwa ada perbedaan sintaksis antara sintaksis asli Maxima dan sintaksis default ekspresi simbolik di EMT.
Matematika simbolik diintegrasikan 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 bilangan yang sangat besar.

```
>$&44!
```

Dengan cara ini, Anda dapat menghitung hasil yang besar dengan tepat. Mari kita menghitung

$$C(44,10)=\frac{44!}{34!\cdot 10!}$$

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

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

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

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali padanya. Misalnya, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima yang disediakan oleh penulis program tersebut.
Anda akan mengetahui bahwa cara berikut juga bisa dilakukan.

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

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

If you want to replace x with any specific value use "with".
Jika Anda ingin mengganti x dengan nilai tertentu, gunakan "with".

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

Dengan begitu Anda bisa menggunakan solusi suatu persamaan di persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasannya adalah adanya tanda simbolis khusus pada string tersebut.

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

Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan $ di depan & (atau Anda dapat menghilangkan &) sebelum perintah. Jangan jalankan perintah Maxima dengan $, jika Anda belum menginstal LaTeX.

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

Ekspresi simbolik diurai oleh Euler. Jika Anda memerlukan sintaksis kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi tersebut di "...". Menggunakan lebih dari sekadar ekspresi sederhana bisa saja dilakukan, namun sangat tidak disarankan.

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

```
25
```

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

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

Sekali lagi, % mengacu pada hasil sebelumnya.

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

```
2432902008176640000
```

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

```
 8  4  2
2  3  5  7
```

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

```
 18  8  4  2
2   3  5  7  11 13 17 19
```

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

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

```
 2
g
```

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

```
 3  x
x  E
```

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

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

```
     5
125 E
```

```
18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

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

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

```
              10          5
        1000 E    - 125 E


        2.20079141499189e+7
```

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

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

```
>tex(fx)
```

```
x^3\,e^{x}
```

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

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

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

Penugasannya juga bisa bersifat simbolis.

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

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

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

Bandingkan dengan perintah numerik "solve" di Euler, yang memerlukan nilai awal, dan opsional nilai target.

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

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membacakan tugas x= dst. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

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

```
[-3.23607,  1.23607]
```

Untuk mendapatkan solusi simbolik tertentu, seseorang dapat menggunakan "dengan" dan indeks.

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

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]
```

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

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

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 simbolik. 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 juga dapat digunakan untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut 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. Daripada ekspresi, kita hanya perlu memberikan nama fungsinya.
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 "timpa". Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah pada fungsi lain yang bergantung pada fungsi tersebut.
Anda masih dapat memanggil fungsi bawaan sebagai "_...", jika fungsi tersebut ada di inti Euler.

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

```
0.707106781187
```

Sebaiknya kita menghilangkan redefinisi dosa ini.

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

```
0.707106781187
```

Parameter Bawaan

Fungsi numerika dapat memiliki parameter default.

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

Menghilangkan parameter ini akan menggunakan nilai default.

```
>f(4)
```

```
16
```

Menyetelnya akan menimpa nilai default.

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

```
80
```

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

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

```
16
```

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

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

```
24
```

Namun parameter yang ditetapkan mengesampingkan nilai global.
Jika argumen tidak ada dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut harus dideklarasikan dengan ":="!

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

```
20
```

Fungsi simbolik didefinisikan dengan "&=". Mereka didefinisikan di Euler dan Maxima, dan bekerja di kedua dunia. Ekspresi yang menentukan dijalankan melalui Maxima sebelum definisi.

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

Mereka juga dapat digunakan dalam ekspresi numerik. Tentu saja, ini hanya akan berfungsi jika EMT dapat menafsirkan 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
>solve(&g(x),0.5)
```

```
0.703467422498
```

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

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

```
0.703467422498
```

```
>function P(x,n) &= (2*x-1)^n; $&P(x,n)
>function Q(x,n) &= (x+2)^n; $&Q(x,n)
>$&P(x,4), $&expand(%)
>P(3,4)
```

```
Function P not found.
Try list ... to find functions!
Error in:
P(3,4) ...
      ^
```

```
>$&P(x,4)+ Q(x,3), $&expand(%)
>$&P(x,4)-Q(x,3), $&expand(%), $&factor(%)
>$&P(x,4)*Q(x,3), $&expand(%), $&factor(%)
>$&P(x,4)/Q(x,1), $&expand(%), $&factor(%)
>function f(x) &= x^3-x; $&f(x)
```

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

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

Dengan := fungsinya numerik. Contoh yang baik adalah integral tertentu

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

yang tidak dapat dievaluasi secara simbolis.
Jika kita mendefinisikan ulang fungsi tersebut dengan kata kunci "peta" maka dapat digunakan untuk vektor x. Secara internal, fungsi ini 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 fungsinya bisa dipanggil dengan atau tanpa parameter "base".

```
>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. Hal ini dimungkinkan dengan parameter vektor.

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

Fungsi simbolik seperti ini dapat digunakan untuk variabel simbolik.
Namun fungsinya juga dapat digunakan untuk vektor numerik.

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

```
17
```

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

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

```
diff(expr, y, 2) + diff(expr, x, 2)
```

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

Namun 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)
```

Untuk meringkas
- &= mendefinisikan fungsi simbolik,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi simbolik murni.

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.
Untuk menyelesaikan ekspresi sederhana dari satu variabel, kita dapat menggunakan fungsi solve(). Dibutuhkan nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode secant.

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

```
1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolik. Ambil fungsi berikut.

```
>$&solve(x^2=2,x)
>$&solve(x^2-2,x)
>$&solve(a*x^2+b*x+c=0,x)
>$&solve([a*x+b*y=c,d*x+e*y=f],[x,y])
>px &= 4*x^8+x^7-x^4-x; $&px
```

Sekarang kita mencari titik yang 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 simbolik dalam bentuk simbolik mengembalikan daftar solusi. Kami menggunakan pemecah simbolis solve() yang disediakan oleh Maxima.

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

Cara termudah untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusi secara numerik seperti halnya 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])
```

Solving systems of equations symbolically can be done with vectors of equations and the symbolic solver solve(). The answer is a list of lists of equations.
Penyelesaian 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])
```

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

$$a^x - x^a = 0.1$$

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 parameternya (cara lainnya adalah parameter titik koma).

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

```
2.54116291558
```

Ini juga berfungsi dengan ekspresi. Namun 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)
```

```
2.54116291558
```

Menyelesaikan Ketidaksamaan

Untuk menyelesaikan ketidaksamaan, EMT tidak akan dapat melakukannya, melainkan dengan bantuan Maxima, artinya secara eksak (simbolik). Perintah Maxima yang digunakan adalah fourier_elim(), yang harus dipanggil dengan perintah "load(fourier_elim)" terlebih dahulu.

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

```
        C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\
ourier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([x^2 - 1>0],[x]) // x^2-1 > 0
>$&fourier_elim([x^2 - 1<0],[x]) // x^2-1 < 0
>$&fourier_elim([x^2 - 1 # 0],[x]) // x^-1 <> 0
>$&fourier_elim([x # 6],[x])
>$&fourier_elim([x < 1, x > 1],[x]) // tidak memiliki penyelesaian
>$&fourier_elim([minf < x, x < inf],[x]) // solusinya R
>$&fourier_elim([x^3 - 1 > 0],[x])
>$&fourier_elim([cos(x) < 1/2],[x]) // ??? gagal
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y]) // sistem pertidaksamaan
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[y,x])
>$&fourier_elim((x + y < 5) and (x - y >8),[x,y])
>$&fourier_elim(((x + y < 5) and x < 1) or  (x - y >8),[x,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])
```

Bahasa Matriks

Dokumentasi inti EMT berisi pembahasan rinci 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
```

Hasil kali 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 semua fungsi dan operator bekerja elemen demi 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 // hasilkali invers A dan b, A^(-1)b
```

```
             -2
            2.5
```

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

```
             -2
            2.5
```

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

```
              1              0
              0              1
```

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

```
              1              0
              0              1
```

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

```
              1              4
              9             16
```

Ini bukan hasil kali matriks, melainkan perkalian elemen demi elemen. Hal yang sama juga berlaku untuk vektor.

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

```
              9
             16
```

Jika salah satu operan adalah vektor atau skalar, maka operan tersebut 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 ini adalah vektor baris, maka diterapkan ke semua kolom A.

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

```
              2             6
              6            12
```

Kita dapat membayangkan perkalian ini seolah-olah vektor baris v telah diduplikasi untuk membentuk matriks yang berukuran 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
```

Hal ini juga berlaku untuk dua vektor dimana yang satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kita menghitung i*j untuk i,j dari 1 sampai 5. Caranya adalah dengan mengalikan 1:5 dengan transposenya. 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, ingatlah 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,  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.
Kita mendapatkan vektor 0 dan 1, dimana 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,  0]
```

Dari vektor tersebut, "bukan nol" memilih elemen bukan nol.
Dalam hal ini, kita 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 bilangan 1 sampai 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. Ia menggunakan floating point presisi ganda secara internal. Namun, seringkali hal ini sangat berguna.
Kita dapat memeriksa primalitasnya. Mari kita cari tahu, berapa banyak persegi ditambah 1 yang merupakan bilangan prima.

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

```
  112
```

Fungsi nonzeros() 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
```

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

Indeks ini dapat digunakan untuk mengatur elemen ke nilai tertentu.

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

```
       0.765761      0.401188      0.406347             0
              0             0      0.495975      0.952814
       0.548138             0      0.444255      0.539246
```

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

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

```
       0.765761      0.401188      0.406347     -0.126917
      -0.122404     -0.691673      0.495975      0.952814
       0.548138     -0.483902      0.444255      0.539246
```

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

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

```
  [0.267829,  0.13673,  0.390567,  0.006085]
```

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]
```

Namun 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
  [-0.765761,  -0.952814,  -0.548138]
```

Fungsi Matriks Lainnya (Matriks Bangunan)

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

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

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

Demikian pula, kita dapat melampirkan matriks ke matriks lain secara berdampingan, jika keduanya mempunyai 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 jumlah barisnya tidak sama, matriks yang lebih pendek diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan real yang melekat pada suatu 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']
```

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

Tujuan utamanya adalah untuk menafsirkan ekspresi vektor untuk vektor kolom.

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

```
            1             1
            2             4
            3             9
```

Untuk mendapatkan ukuran A, kita bisa 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
```

Masih banyak fungsi lain yang menghasilkan matriks.

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

```
            1             1
            1             1
```

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

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

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

Matriks bilangan acak juga dapat dihasilkan dengan acak (distribusi seragam) atau normal (distribusi Gauß).

```
>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 elemen2 1, 2, 3, ..., 9 ke bentuk matriks 3x3
```

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

Dengan fungsi berikut, kita dapat menggunakan fungsi ini dan fungsi dup untuk menulis fungsi rep(), yang mengulangi vektor sebanyak 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,  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,  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]
```

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 perlu mencari elemennya terlebih dahulu. Fungsi indexof(v,x) dapat digunakan untuk mencari elemen x dalam vektor yang diurutkan 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 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 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 mendemonstrasikannya, kami menyusun ulang 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 diagonalnya.

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

```
  [1,   5,   9]
```

Misalnya. Kita dapat membagi matriks dengan diagonalnya. Bahasa matriks menjaga agar vektor kolom d diterapkan pada matriks baris demi baris.

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

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

Vektorisasi

Hampir semua fungsi di Euler juga berfungsi untuk input matriks dan vektor, jika hal 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 ditampikan
```

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

$$s = t^3 - t$$

```
>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 dikali vektor baris diperluas ke matriks, jika operator diterapkan. Berikut ini, v' adalah vektor yang dialihkan (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, ini sangat berbeda dengan perkalian matriks. Hasil kali matriks dilambangkan dengan titik "." di EMT.

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

```
55
```

Secara default, vektor baris dicetak dalam format ringkas.

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

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

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

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

```
5
25
```

Untuk mengubah urutan matriks kita menggunakan apostrof.

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

```
              1
              2
              3
              4
```

Jadi kita dapat menghitung matriks A dikalikan 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 dengan 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 berfungsi seperti bilangan real.

```
>v.v'
```

```
  30
```

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

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

```
  30
```

Operator dan fungsi mematuhi bahasa matriks Euler.
Berikut ringkasan peraturannya.
- Suatu fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.
- Operator yang mengoperasikan dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemen-elemen matriks.
- Jika kedua matriks mempunyai dimensi yang berbeda, keduanya diekspansi secara wajar sehingga mempunyai ukuran yang sama.
Misalnya, nilai skalar dikalikan vektor dengan mengalikan nilai setiap elemen vektor. Atau matriks dikalikan vektor (dengan *, bukan .) memperluas vektor ke ukuran matriks dengan menduplikasinya.
Berikut ini adalah kasus sederhana dengan operator ^.

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

```
  [1,  4,  9]
```

Ini kasus yang lebih rumit. Vektor baris dikalikan vektor kolom memperluas keduanya dengan cara menduplikasi.

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

```
               1             2             3
               2             4             6
               3             6             9
```

Perhatikan bahwa perkalian skalar menggunakan perkalian matriks, bukan *!

```
>v.v'
```

```
14
```

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

```
sum,prod menghitung jumlah dan hasil kali baris
cumsum,cumprod melakukan hal yang sama secara kumulatif
menghitung nilai ekstrem setiap baris
extreme mengembalikan vektor dengan informasi ekstrem
diag(A,i) mengembalikan diagonal ke-i
setdiag(A,i,v) menyetel diagonal ke-i
id(n) matriks identitas
det(A) determinannya
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 dengan 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 terdapat operator "|" Dan "_".

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

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

Elemen-elemen 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 tersebut. Untuk matriks, ini mengembalikan baris ke-i yang lengkap dari matriks tersebut.

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

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

Indeks juga dapat berupa vektor baris dari indeks. : menunjukkan semua indeks.

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

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

Bentuk kependekan dari : menghilangkan indeks sepenuhnya.

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

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

Untuk tujuan vektorisasi, elemen matriks dapat diakses seolah-olah elemen tersebut 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 pada tabel, mari kita atur ulang ke format default, dan hitung tabel nilai sinus dan kosinus. Perhatikan bahwa sudut dinyatakan 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 sekaligus.

Dalam contoh berikut, kita menghitung t[j]^i untuk i dari 1 hingga n. Kita mendapatkan sebuah matriks, yang setiap barisnya merupakan tabel t^i untuk satu i. Yaitu, matriks memiliki elemen lateks: a_{i,j} = t_j^i, \quad 1 \le j \le 101, \quad 1 \le i \le n

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

Integrasi numerik integral() hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektorisasinya.

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

Kata kunci "peta" membuat vektorisasi fungsi tersebut. Fungsinya sekarang akan berfungsi untuk vektor bilangan.

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

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

Sub-Matriks dan Elemen Matriks

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 baris matriks secara 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 1xn dan mxn, tentukan semua kolom menggunakan indeks kedua yang kosong.

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

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

Jika indeks adalah vektor dari indeks, Euler akan mengembalikan baris matriks yang sesuai.
Di sini kita menginginkan baris pertama dan kedua A.

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

```
             1             2             3
             4             5             6
```

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

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

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

Trik indeks juga berfungsi dengan kolom.
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
```

Alternatively, leave the first index empty.
Alternatifnya, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir 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
```

Kita juga dapat memberikan nilai pada baris A.

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

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

Kita bahkan dapat menetapkan sub-matriks jika ukurannya sesuai.

```
>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, bergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Namun perlu diingat bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

```
>A[4]
```

```
Row 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 mengetahui indeks vektor yang diurutkan dalam vektor aslinya. Ini dapat digunakan untuk menyusun ulang vektor lain dengan cara yang sama.
Mari kita mengacak sebuah vektor.

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

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

Indeks berisi urutan v.

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

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

Ini juga berfungsi untuk vektor string.

```
>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 juga berfungsi untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar linier

EMT memiliki banyak sekali fungsi untuk menyelesaikan masalah sistem linier, sistem sparse, atau regresi. Untuk sistem linier Ax=b, Anda dapat menggunakan algoritma Gauss, matriks invers, atau linear fit. Operator A\b menggunakan versi algoritma Gauss.

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

```
             -4
            4.5
```

Contoh lain, kita membuat matriks berukuran 200x200 dan jumlah baris-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 tepat.

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

```
  8.790745908981989e-13
```

Jika sistem tidak mempunyai 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
```

Penentu matriks ini adalah 0.

```
>det(A)
```

```
0
```

Matriks Simbolik

Maxima memiliki matriks simbolik. Tentu saja Maxima dapat digunakan untuk permasalahan aljabar linier sederhana seperti itu. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, lalu menggunakannya dalam ekspresi simbolik. Bentuk [...] yang biasa untuk mendefinisikan matriks dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
>$&det(A), $&factor(%)
>$&invert(A) with a=0
>A &= [1,a;b,2]; $A
```

Seperti semua variabel simbolik, matriks ini dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&det(A-x*ident(2)), $&solve(%,x)
```

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah sebuah vektor dengan dua vektor nilai eigen dan multiplisitas.

```
>$&eigenvalues([a,1;1,a])
```

Untuk mengekstrak vektor eigen tertentu memerlukan pengindeksan yang cermat.

```
>$&eigenvectors([a,1;1,a]), &%[2][1][1]
```

```
[1, - 1]
```

Matriks simbolik dapat dievaluasi dalam Euler secara numerik sama seperti ekspresi simbolik lainnya.

```
>A(a=4,b=5)
```

```
            1             4
            5             2
```

Dalam ekspresi simbolik, gunakan dengan.

```
>$&A with [a=4,b=5]
```

Akses ke deretan matriks simbolik berfungsi sama seperti matriks numerik.

```
>$&A[1]
```

Ekspresi simbolis dapat berisi tugas. Dan itu mengubah matriks A.

```
>&A[1,1]:=t+1; $&A
```

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
>B &:= [1,2;3,4]; $B, $&invert(B)
```

Hasilnya dapat dievaluasi secara numerik dalam Euler. Untuk informasi lebih lanjut tentang Maxima, lihat pengenalan Maxima.

```
>$&invert(B)()
```

```
             -2              1
            1.5           -0.5
```

Euler juga memiliki fungsi kuat xinv(), yang melakukan upaya lebih besar dan mendapatkan hasil yang lebih tepat.

Perhatikan, bahwa dengan &:= matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan numerik dalam ekspresi numerik. Jadi kita bisa menggunakannya di sini.

```
>longest B.xinv(B)
```

```
                        1                        0
                        0                        1
```

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)
```

Nilai Numerik dalam Ekspresi simbolik

Ekspresi simbolis hanyalah string yang berisi ekspresi. Jika kita ingin mendefinisikan nilai untuk ekspresi simbolik dan ekspresi numerik, kita harus menggunakan "&:=".

```
>A &:= [1,pi;4,5]
```

```
              1        3.14159
              4              5
```

Masih terdapat perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolik, pendekatan pecahan untuk real akan digunakan.

```
>$&A
```

Untuk menghindari hal ini, ada fungsi "mxmset(variabel)".

```
>mxmset(A); $&A
```

Maxima juga dapat menghitung dengan bilangan floating point, bahkan dengan bilangan mengambang besar dengan 32 digit. Namun evaluasinya jauh lebih lambat.

```
>$&bfloat(sqrt(2)), $&float(sqrt(2))
```

Ketepatan angka floating point besar dapat diubah.

```
>&fpprec:=100; &bfloat(pi)
```

```
        3.14159265358979323846264338327950288419716939937510582097494\
459230781640628620899862803482534211706 8b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolik 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)
```

Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk menghitung 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 sebesar 5.000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

Sekarang kami mengasumsikan tingkat bunga 3% per tahun. Mari kita tambahkan satu tarif sederhana dan hitung hasilnya.

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaks berikut.

```
>K+K*3%
```

```
5150
```

Namun lebih mudah menggunakan faktor tersebut

```
>q=1+3%, K*q
```

```
1.03
5150
```

Selama 10 tahun, kita cukup mengalikan faktor-faktornya dan mendapatkan nilai akhir dengan tingkat bunga majemuk.

```
>K*q^10
```

```
6719.58189672
```

Untuk keperluan kita, kita dapat mengatur formatnya menjadi 2 digit setelah titik desimal.

```
>format(12,2); K*q^10
```

```
    6719.58
```

Mari kita cetak yang dibulatkan menjadi 2 digit dalam satu 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 ke 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis satu perulangan, tetapi cukup masuk

```
>K*q^(0:10)
```

```
Real 1 x 11 matrix

    5000.00     5150.00     5304.50     5463.64     ...
```

Bagaimana keajaiban ini terjadi? 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 di Euler dapat diterapkan pada vektor elemen demi 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 kita mendapatkan vektor nilainya.

```
>VK=K*q^(0:10);
```

Tentu saja, cara realistis untuk menghitung suku bunga ini adalah dengan membulatkan ke sen terdekat setiap tahunnya. Mari kita tambahkan fungsi untuk ini.

```
>function oneyear (K) := round(K*q,2)
```

Mari kita bandingkan kedua hasil tersebut, 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 mengulanginya selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.
Cara termudah adalah fungsi iterate, yang mengulangi fungsi tertentu beberapa kali.

```
>VKr=iterate("oneyear",5000,10)
```

```
  Real 1 x 11 matrix

      5000.00     5150.00     5304.50     5463.64     ...
```

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 vektor tertentu, 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. Ingatlah 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 tahunnya.

```
>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)
```

```
  Real 1 x 11 matrix

      5000.00     5350.00     5710.50     6081.82     ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",5000,10)
```

```
  Real 1 x 11 matrix

      5000.00     4950.00     4898.50     4845.45     ...
```

Kami melihat uangnya berkurang. Jelasnya, jika kita hanya mendapat bunga sebesar 150 pada tahun pertama, namun menghapus 200, kita kehilangan uang setiap tahunnya.
Bagaimana kita dapat menentukan berapa tahun uang tersebut akan bertahan? Kita harus menulis satu lingkaran untuk ini. Cara termudah adalah dengan melakukan iterasi cukup lama.

```
>VKR=iterate("onepay",5000,50)
```

```
  Real 1 x 51 matrix

      5000.00     4950.00     4898.50     4845.45     ...
```

Dengan menggunakan bahasa matriks, kita dapat menentukan nilai negatif pertama dengan cara berikut.

```
>min(nonzeros(VKR<0))
```

```
        48.00
```

Alasannya adalah bukan nol (VKR<0) mengembalikan vektor indeks i, dengan VKR[i]<0, dan min menghitung indeks minimal.

Karena vektor selalu dimulai dengan indeks 1, maka jawabannya adalah 47 tahun.

Fungsi iterate() memiliki satu trik lagi. Ini dapat 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 mengetahui bahwa nilainya adalah 0 setelah 50 tahun. Berapa tingkat bunganya?

Ini adalah pertanyaan yang hanya bisa dijawab secara numerik. Di bawah ini, kita akan mendapatkan rumus yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada rumus yang mudah untuk menentukan tingkat suku bunga. Namun untuk saat ini, kami menargetkan 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

lateks: x_{n+1} = x_n \cdot \kiri(1+ \frac{P}{100}\kanan) + R

Namun 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.

Apalagi 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)
```

```
3.15
```

Rutinitas penyelesaian menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami mengambil nilai awal 3% untuk algoritma. Fungsi 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 3% per tahun.

```
>solve("f(5000,x,3,20)",-200)
```

```
-336.08
```

Perhatikan bahwa Anda tidak dapat menyelesaikan jumlah tahun, karena fungsi kami mengasumsikan n sebagai nilai bilangan bulat.

Solusi Simbolis Masalah Suku Bunga

Kita dapat menggunakan bagian simbolis dari Euler untuk mempelajari masalahnya. Pertama kita mendefinisikan fungsi onepay() kita secara simbolis.

```
>function op(K) &= K*q+R; $&op(K)
```

Sekarang kita dapat mengulanginya.

```
>$&op(op(op(op(K)))), $&expand(%)
```

Kami melihat sebuah pola. Setelah n periode yang kita miliki

$$K_n = q^n K + R(1 + q + \ldots + q^{n-1}) = q^n K + \frac{q^n - 1}{q - 1} R$$

Rumusnya adalah rumus jumlah geometri yang diketahui Maxima.

```
>&sum(q^k,k,0,n-1); $& % = ev(%,simpsum)
```

Ini agak rumit. Jumlahnya dievaluasi dengan tanda "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)
```

Fungsinya sama dengan fungsi f kita sebelumnya. Tapi ini lebih efektif.

```
>longest f(5000,-200,3,47), longest fs(5000,-200,3,47)
```

```
      -19.82504734650985
      -19.82504734652684
```

Sekarang kita dapat menggunakannya untuk menanyakan waktu n. Kapan modal kita habis? Perkiraan awal kami adalah 30 tahun.

```
>solve("fs(5000,-330,3,x)",30)
```

```
       20.51
```

Jawaban ini mengatakan akan menjadi negatif setelah 21 tahun.
Kita juga dapat menggunakan sisi simbolis Euler untuk menghitung rumus 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). Rumusnya jelas

```
>equ &= fs(K,R,P,n)=Kn; $&equ
```

Biasanya rumus ini diberikan dalam bentuk

$$i = \frac{P}{100}$$

```
>equ &= (equ with P=100*i); $&equ
```

Kita dapat menyelesaikan nilai R secara simbolis.

```
>$&solve(equ,R)
```

Seperti yang Anda lihat dari rumusnya, fungsi ini mengembalikan kesalahan floating point untuk i=0. Euler tetap merencanakannya.
Tentu saja, kami memiliki batasan berikut.

```
>$&limit(R(5000,0,x,10),x,0)
```

Yang jelas tanpa bunga kita harus membayar kembali 10 tarif 500.
Persamaan tersebut juga dapat diselesaikan untuk n. Akan terlihat lebih bagus jika kita menerapkan beberapa penyederhanaan padanya.

```
>fn &= solve(equ,n) | ratsimp; $&fn
```

Menjawab Soal-Soal Aljabar

1. Sederhanakan bentuk eksponen berikut

$$\left(\frac{24a^{10}b^{-8}c^{7}}{12a^{6}b^{-3}c^{5}}\right)^{-5}$$

penyelesaian:

```
>$&((24*a^10*b^-8*c^7)/(12*a^6*b^-3*c^5))^-5
```

2. Sederhanakan bentuk eksponen berikut

$$\left(\frac{125p^{12}q^{-14}r^{22}}{25p^{8}q^{6}r^{-15}}\right)^{-4}$$

Penyelesaian:

```
>$&((125*p^12*q^-14*r^22)/(25*p^8*q^6*r^-15))^-4
```

3. Hitunglah

$$2^6 \cdot 2^{-3} \div 2^{10} \div 2^{-8}$$

Penyelesaian:

```
>$&(2^6*2^-3/2^10/2^-8)
```

4. 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
```

5. 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
```

Rumus Rencana Tabungan

$$S = P\left[\frac{\left(1 + \frac{r}{12}\right)^{12 \cdot t} - 1}{\frac{r}{12}}\right]$$

dengan
S: Akumulasi jumlah dalam rencana tabungan
P: Setoran/deposit yang dilakukan setiap bulan dalam satuan dollar
t: Lama waktu menabung dalam satuan tahun
r: Tingkat bunga
6. James mendepositkan $250 ke dalam rekening pensiun nya setiap bulan dimulai pada usia 40. Jika investasi menghasilkan bunga sebesar 5%, dimajemukkan setiap bulan, berapa banyak jumlah yang terakumulasi pada rekening ketika ia pensiun 27 tahun kemudian?
Penyelesaian:

$$S = 250\left[\frac{\left(1 + \frac{5/100}{12}\right)^{12 \cdot 27} - 1}{\frac{5/100}{12}}\right]$$

```
>&250*(((1+(0.05/12))^(12*27)-1)/(0.05/12))
```

```
170797.3032920518
```

Jadi, jumlah yang terakumulasi dalam rekening pensiun James 27 tahun kemudian adalah $170797,3032920518
(Soal 7-9) Asumsikan bahwa semua eksponen adalah bilangan bulat, semua penyebutnya bukan nol, dan nol tidak dinaikkan ke pangkat non-positif.
7. Sederhanakan

$$(m^{x-b} \cdot n^{x+b})^x (m^b n^{-b})^x$$

Penyelesaian:

```
>$&(m^(x-b)*n^(x+b))^x*(m^b*n^(-b))^x
```

8. Sederhanakan

$$\left[\frac{(3x^a y^b)^3}{(-3x^a y^b)^2}\right]^2$$

Penyelesaian:

```
>$&(((3*x^a*y^b)^3)/((-3*x^a*y^b)^2))^2
```

9. Sederhanakan

$$\left[\left(\frac{x^r}{y^t}\right)^2 \left(\frac{x^{2r}}{y^{4t}}\right)^{-2}\right]^{-3}$$

penyelesaian:

```
>$&(((x^r)/(y^t))^2*((x^(2*r))/(y^(4*t)))^(-2))^(-3)
```

10. Lakukan operasi yang ditunjukkan

$$(2x + 3y + z - 7) + (4x - 2y - z + 8) + (-3x + y - 2z - 4)$$

Penyelesaian:

```
>&powerdisp:true;
>$&(2*x+3*y+z-7)+(4*x-2*y-z+8)+(-3*x+y-2*z-4)
```

11. Lakukan operasi yang ditunjukkan

$$(5x^2 + 4xy - 3y^2 + 2) - (9x^2 - 4xy + 2y^2 - 1)$$

Penyelesaian:

```
>$&(5*x^2+4*x*y-3*y^2+2)-(9*x^2-4*x*y+2*y^2-1)
```

12. Lakukan operasi yang ditunjukkan

$$(6xy^3)(9x^4y^2)$$

Penyelesaian:

```
>$&(6*x*y^3)*(9*x^4*y^2)
>&powerdisp:false;
```

13. Lakukan operasi yang ditunjukkan

$$(2x^4-3x^2+7x)-(5x^3+2x^2-3x+5)$$

penyelesaian:

```
>$&(2*x^4-3*x^2+7*x)-(5*x^3+2*x^2-3*x+5)
```

14. Lakukan operasi yang ditunjukkan

$$(-5m^4n^2)(6m^2n^3)$$

penyelesaian:

```
>$(-5*m^4*n^2)*(6*m^2*n^3)
```

15. Lakukan operasi yang ditunjukkan

$$(2x^2-3y)^2$$

penyelesaian:

```
>$expand((2*x^2-3*y)^2)
```

16. Lakukan operasi yang ditunjukkan

$$(a-b)(2a^3-ab+3b^2)$$

penyelesaian:

```
>$expand((a-b)*(2*a^3-a*b+3*b^2))
```

17. Lakukan operasi yang ditunjukkan

$$(5x + 2y + 3)(5x + 2y - 3)$$

penyelesaian:

```
>$expand((5*x+2*y+3)*(5*x+2*y-3))
```

18. Lakukan operasi yang ditunjukkan

$$(y - 2)(y + 2)(y^2 + 4)$$

penyelesaian:

```
>$expand((y-2)*(y+2)*(y^2+4))
```

19. 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))
```

20. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$[(2x - 1)^2 - 1]^2$$

penyelesaian:

```
>&powerdisp:false;
>$&expand(((2*x-1)^2-1)^2)
```

21. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(x^{3m} - t^{5n})^2$$

penyelesaian:

```
>$expand((x^(3*m)-t^(5*n))^2)
```

22. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(a + b + c)^2$$

penyelesaian:

```
>&powerdisp:true;
>$expand((a+b+c)^2)
>&powerdisp:false;
```

23. Faktorkan

$$y^2 - 6y + 5$$

penyelesaian:

```
>$&factor(y^2-6*y+5)
```

24. Faktorkan

$$x^2y^2 - 14xy + 49$$

penyelesaian:

```
>$&factor(x^2*y^2-14*x*y+49)
```

25. Faktorkan

$$16a^7b + 54ab^7$$

penyelesaian:

```
>$&factor(16*a^7*b+54*a*b^7)
```

26. Faktorkan

$$25ab^4 - 25az^4$$

penyelesaian:

```
>$&factor(25*a*b^4-25*a*z^4)
```

27. Faktorkan

$$m^3 - 216$$

penyelesaian:

```
>$&factor(m^3-216)
```

28. Faktorkan

$$y^2 - \frac{8}{49} + \frac{2}{7}y$$

penyelesaian:

```
>$&factor(y^2-(8/49)+(2/7)*y)
```

29. Faktorkan

$$(y-4)^2 + 5(y-4) - 24$$

penyelesaian:

```
>$&factor((y-4)^2+5*(y-4)-24)
```

30. Faktorkan

$$(x+h)^3 - x^3$$

penyelesaian:

```
>$&factor((x+h)^3-x^3)
```

31. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$4x^{2n} - 4x^n - 3$$

penyelesaian:

```
>$&factor(4*x^(2*n)-4*x^n-3)
```

32. 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)
```

33. 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))
```

34. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$x^{6a} - t^{3b}$$

penyelesaian:

```
>$&factor(x^(6*a)-t^(3*b))
```

```
35. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan
```

bilangan asli. Faktorkan

```
latex: (y-1)^4 - (y-1)^2
penyelesaian:
```

BAB 2

KB PEKAN 4: MENGGUNAKAN EMT UNTUK MENGAMBAR GRAFIK 2 DIMENSI (2D)

[a4paper,10pt]article eumat

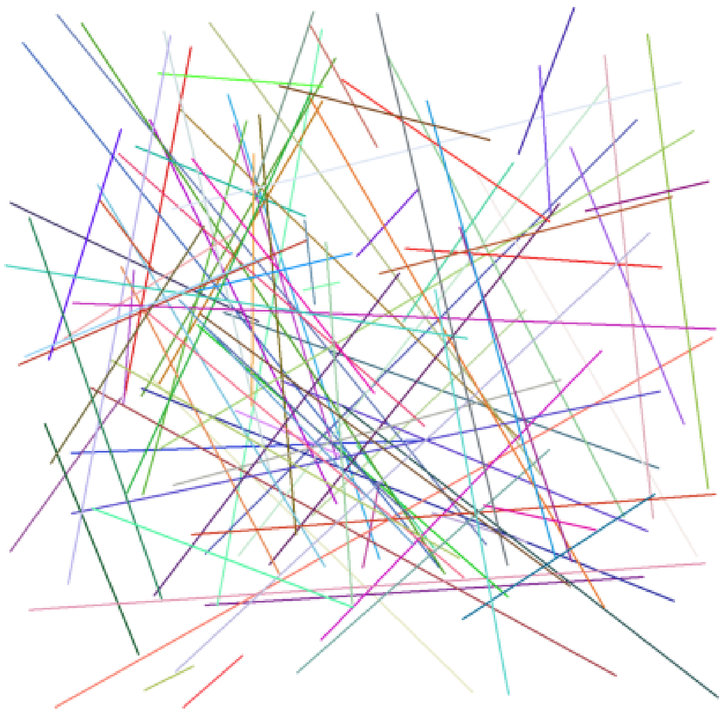
Menggambar Grafik 2D dengan EMT

Nama : Akhmad Rizki Khamid
NIM : 22305141008
Kelas: Matematika E
Notebook ini menjelaskan tentang cara menggambar berbagaikurva 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.

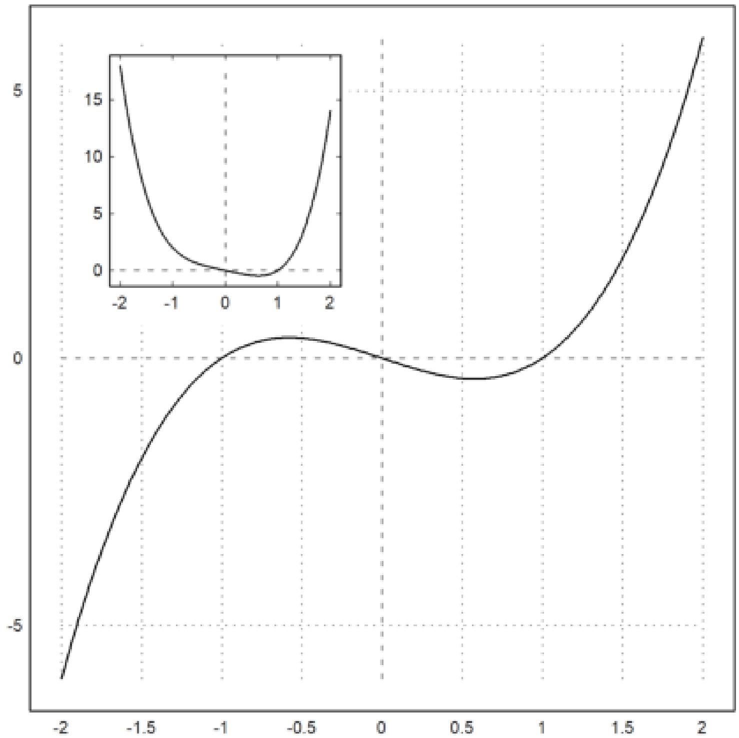
```
>clg; // clear screen
>window(0,0,1024,1024); // use all of the window
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2);  // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]); 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);
>plot2d("x^4-x",grid=6):
```



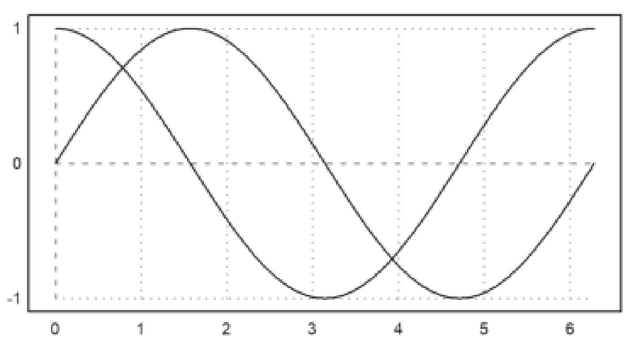

```
>hold off;
>window(ow);
```

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

Aspek 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

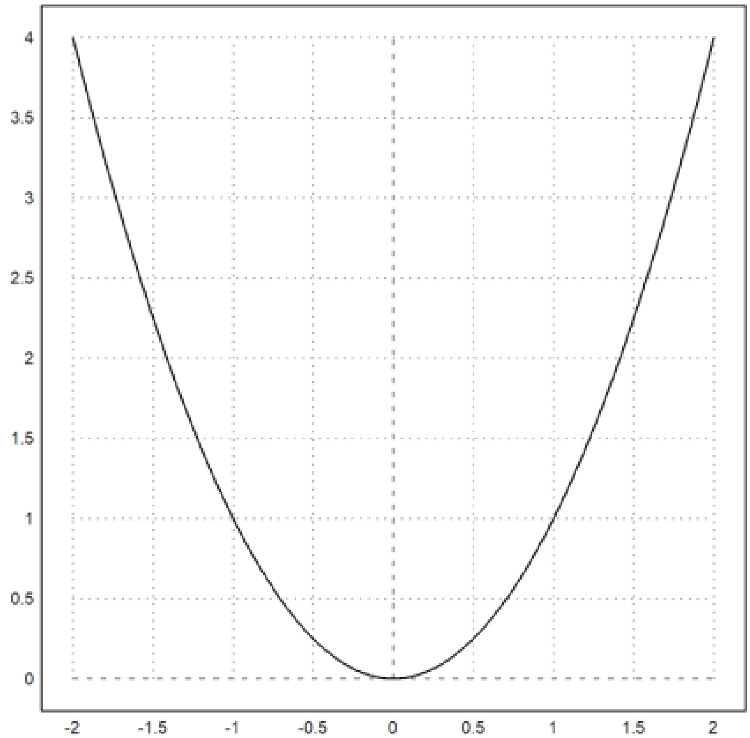
EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat memplot fungsi dan data.
Dimungkinkan untuk membuat plot di Maxima menggunakan Gnuplot atau dengan Python menggunakan Math Plot Lib.
Euler dapat memplot plot 2D dari
- ekspresi
- fungsi, 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 berbayang.

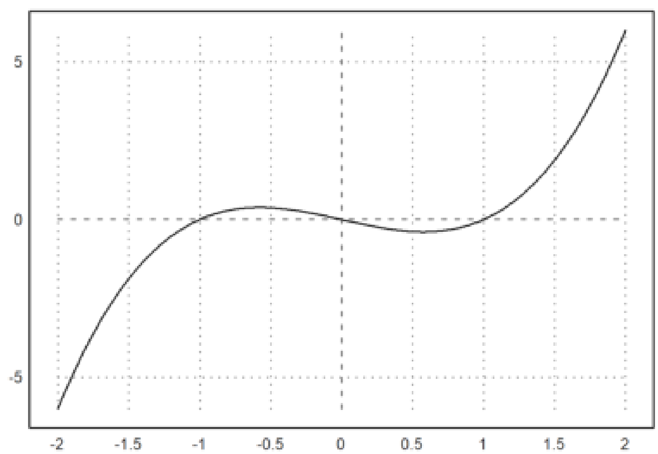
Plot Ekspresi atau Variabel

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.

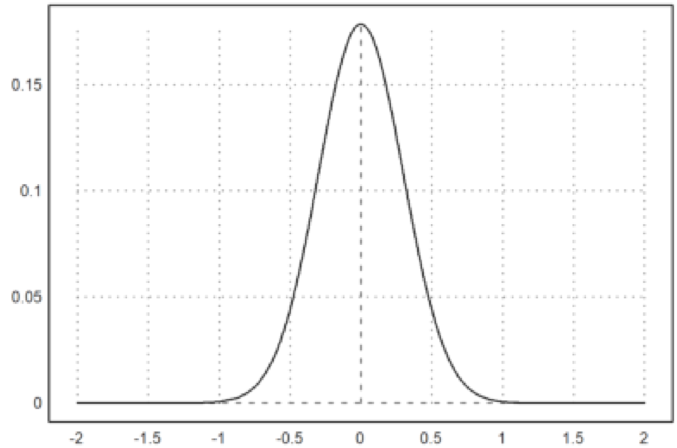
```
>plot2d("x^2"):
```




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

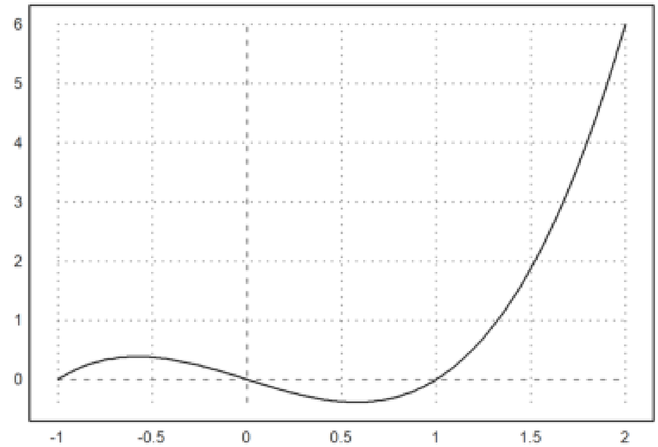



```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```

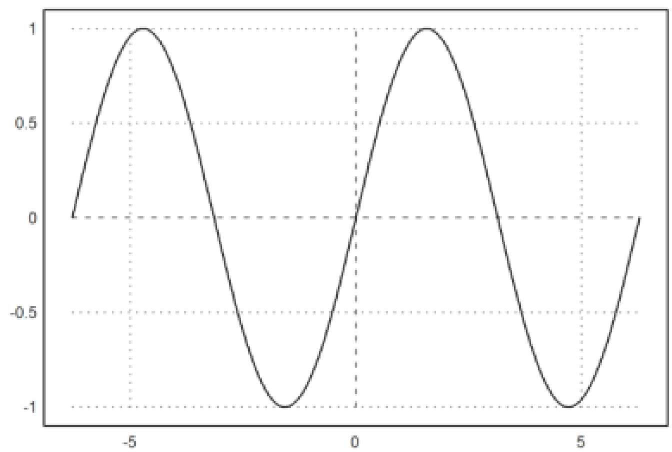


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.
The plot range is set with the following assigned parameters
- a,b: x-range (default -2,2)
- c,d: y-range (default: scale with values)
- r: alternatively a radius around the plot center
- cx,cy: the coordinates of the plot center (default 0,0)

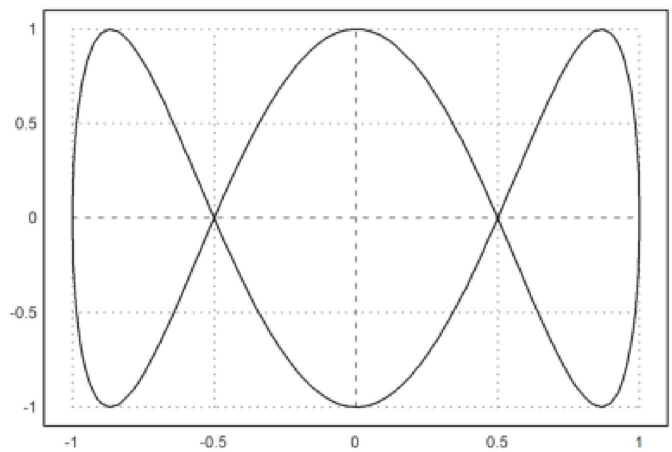
```
>plot2d("x^3-x",-1,2):
```




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



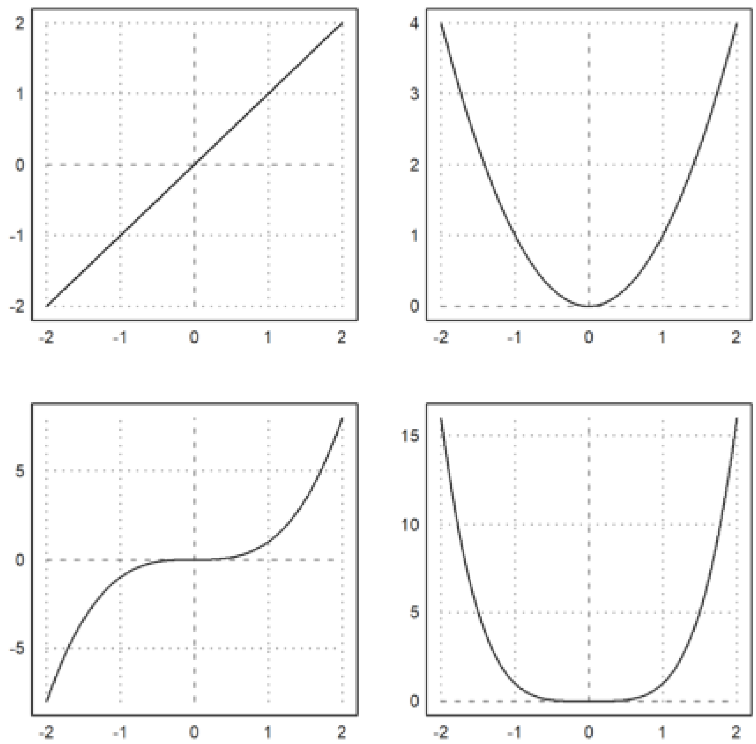
```
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi):
```



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.

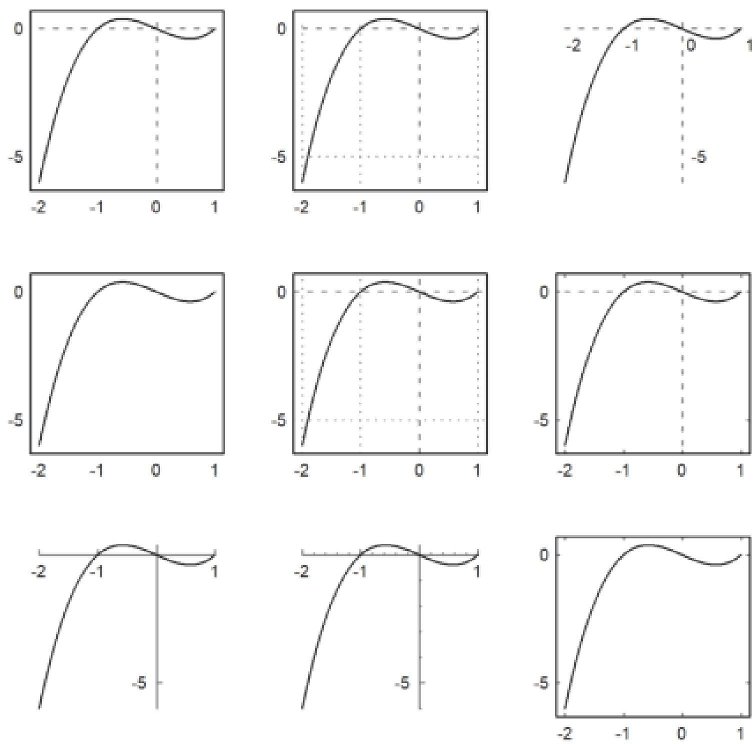
Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.
Bagaimanapun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh, kami memplot x^1 hingga x^4 menjadi 4 bagian jendela. figure(0) mengatur ulang jendela default.

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



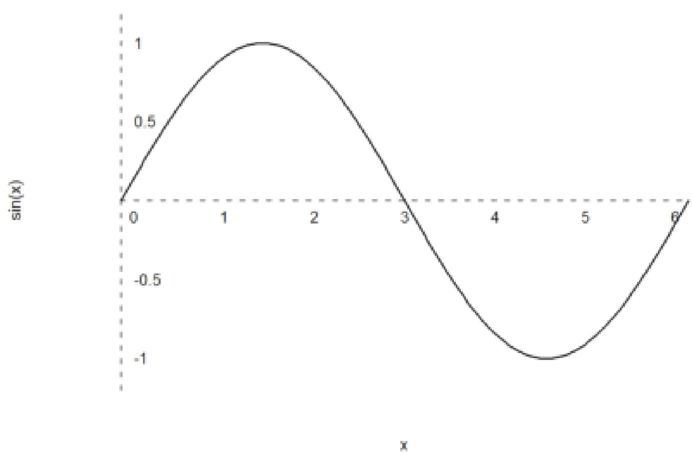
Di plot2d(), ada gaya alternatif yang tersedia dengan grid=x. Untuk gambaran umum, kami menunjukkan berbagai gaya kisi dalam satu gambar (lihat di bawah untuk perintah figure()). Gaya kisi=0 tidak disertakan. Ini menunjukkan tidak ada grid dan tidak ada bingkai.

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

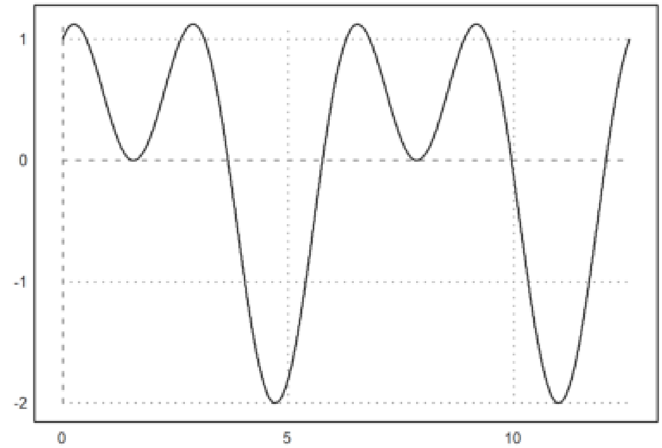


Jika argumen ke plot2d() adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.
Atau, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan sebagai a=... dll.
Dalam contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

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

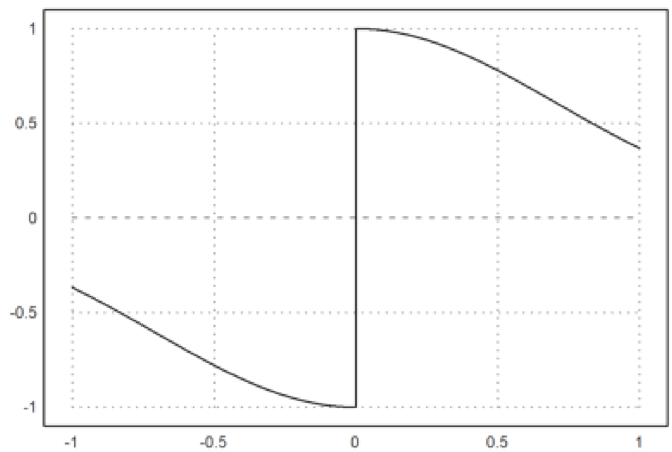



```
>plot2d("sin(x)+cos(2*x)",0,4pi):
```

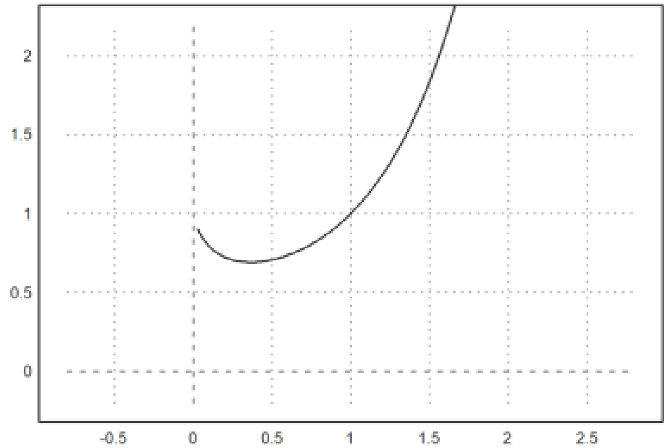


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "gambar". Mereka juga digunakan oleh ekspor HTML.
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, 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):
```

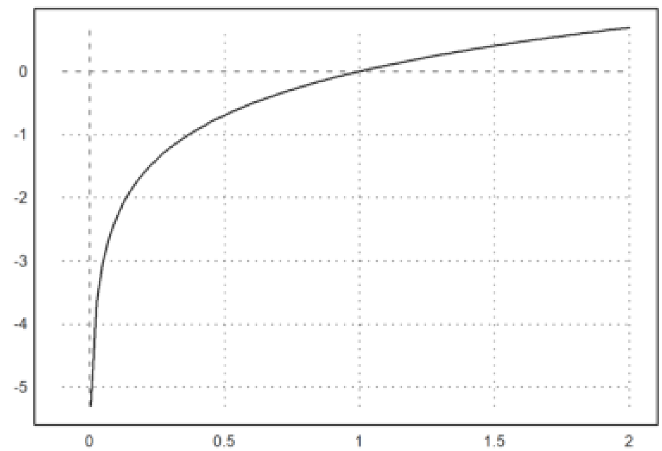



```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



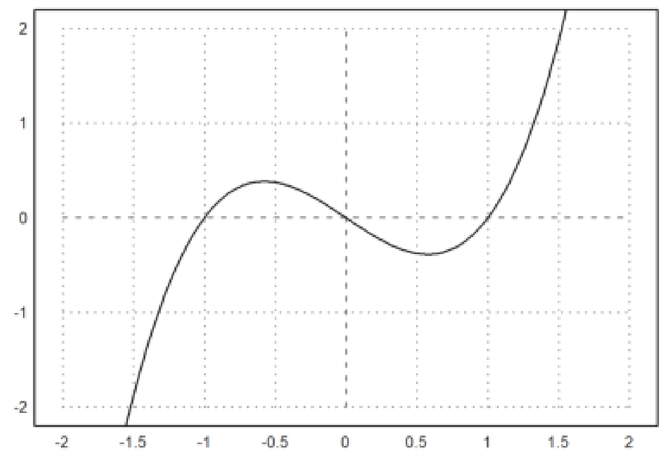
Perhatikan bahwa x^x tidak didefinisikan untuk x<=0. Fungsi plot2d menangkap kesalahan ini, dan mulai merencanakan segera setelah fungsi didefinisikan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN keluar dari jangkauan definisinya.

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

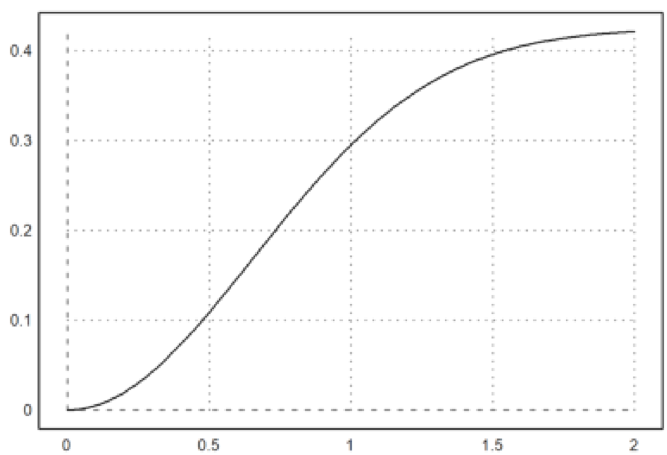


Parameter square=true (atau >square) memilih y-range secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

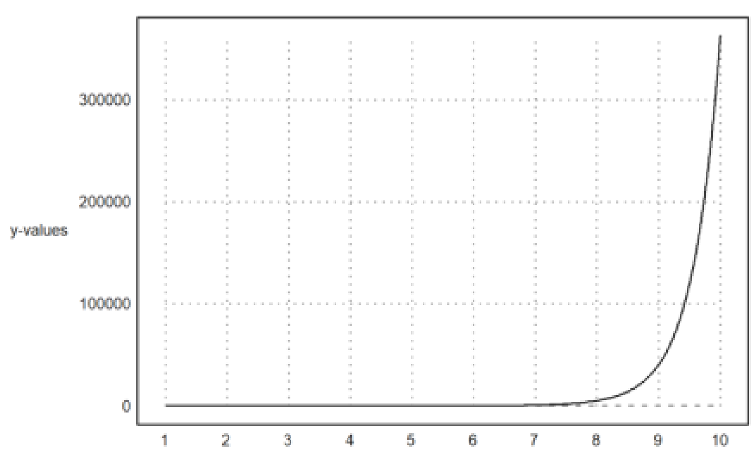



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



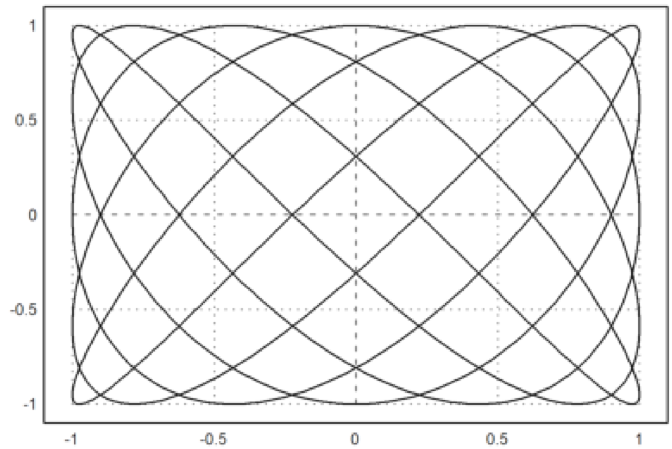
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di plot2d().

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

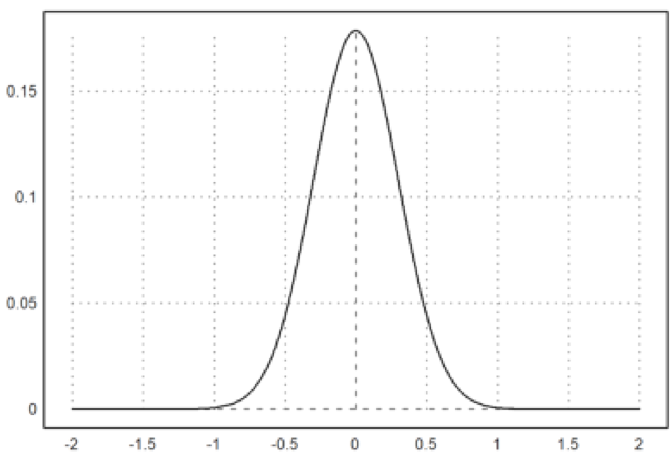


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

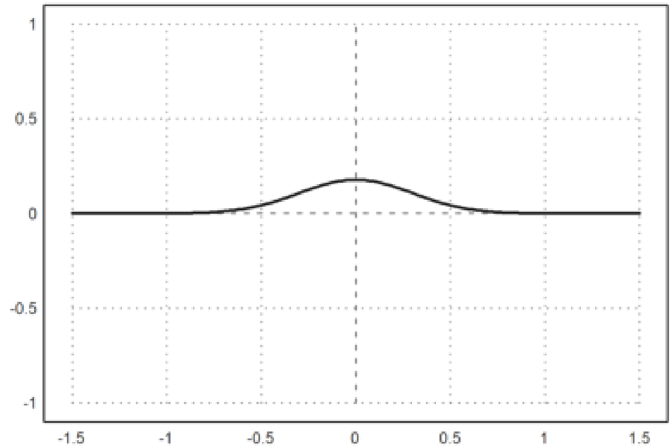
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



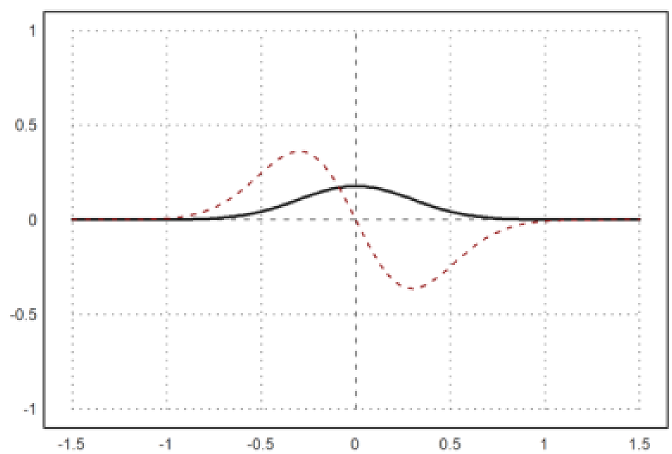

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



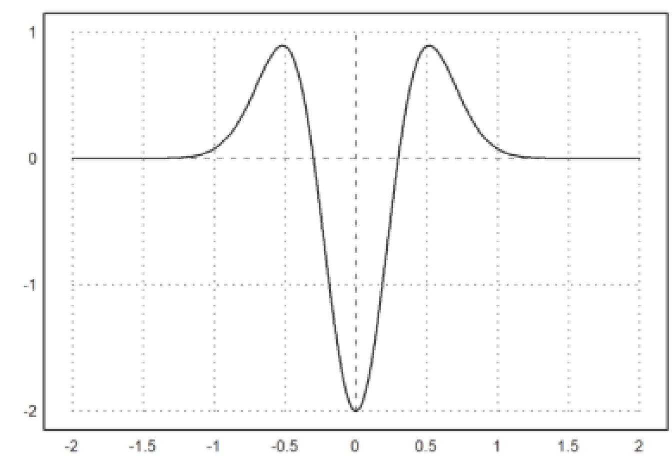

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



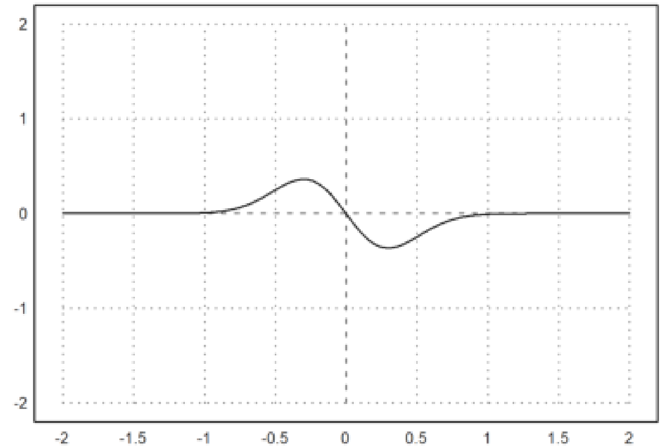
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



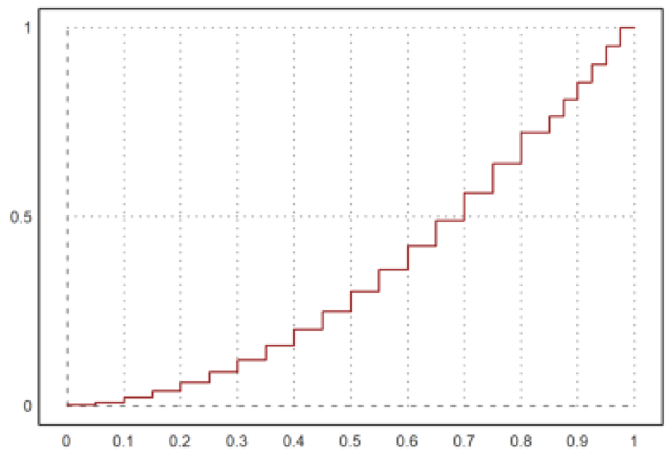
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



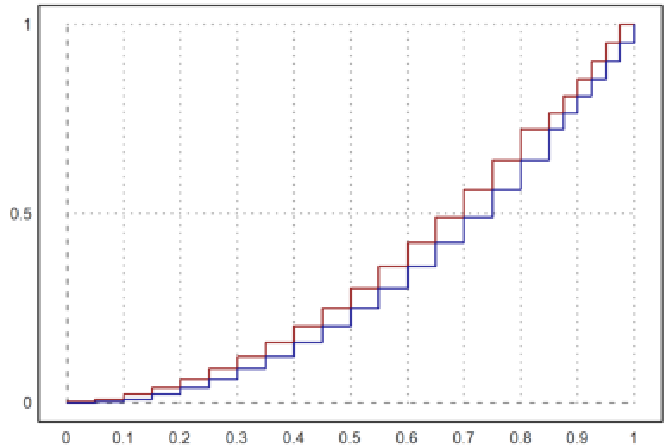
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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



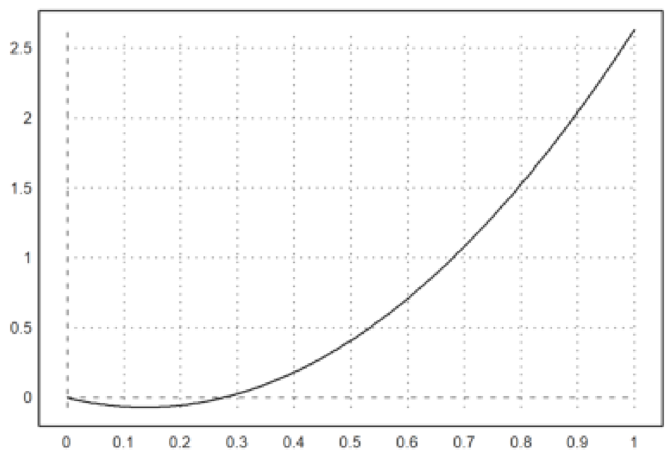

```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



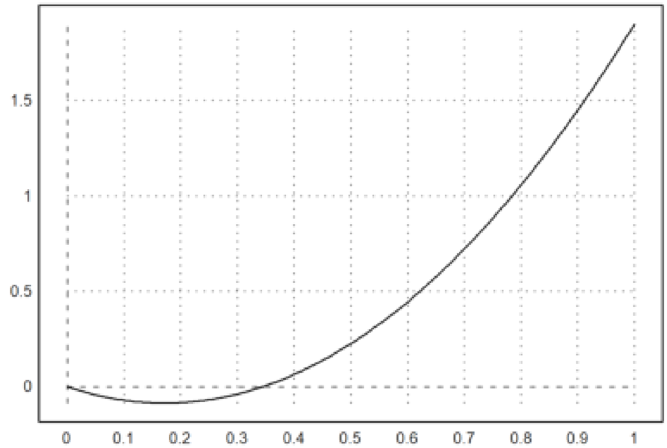
Fungsi dalam satu Parameter

Fungsi plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.
Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

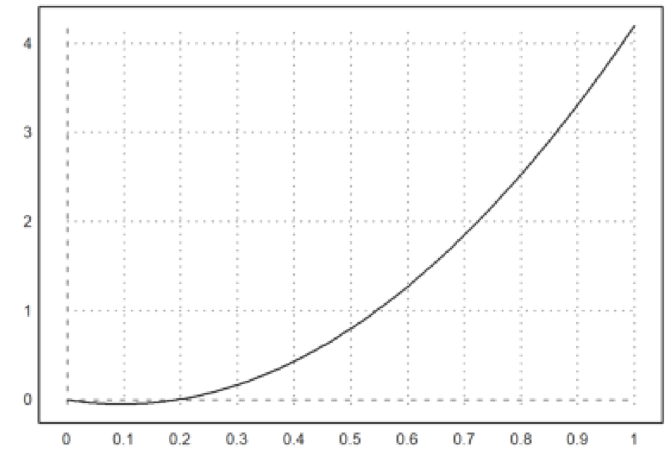
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



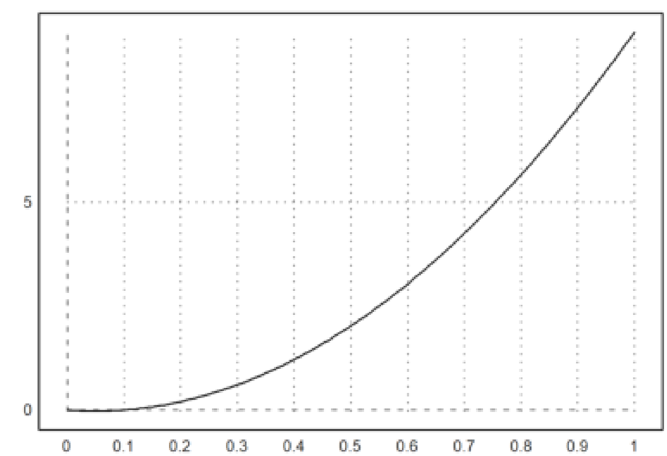
```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



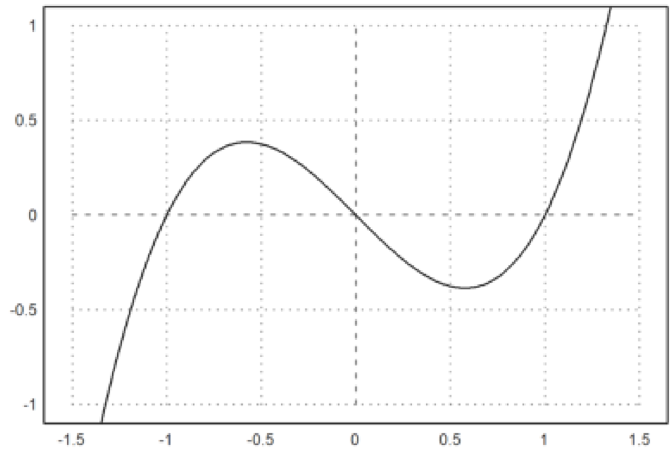

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```




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



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

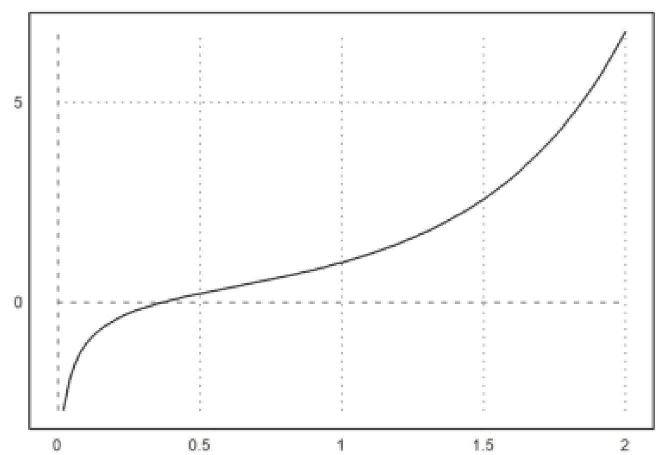


Berikut adalah ringkasan dari fungsi yang diterima
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f
Fungsi plot2d() juga menerima fungsi simbolis. Untuk fungsi simbolis, nama saja yang berfungsi.

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

```
                         x
                        x   (log(x) + 1)
```

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

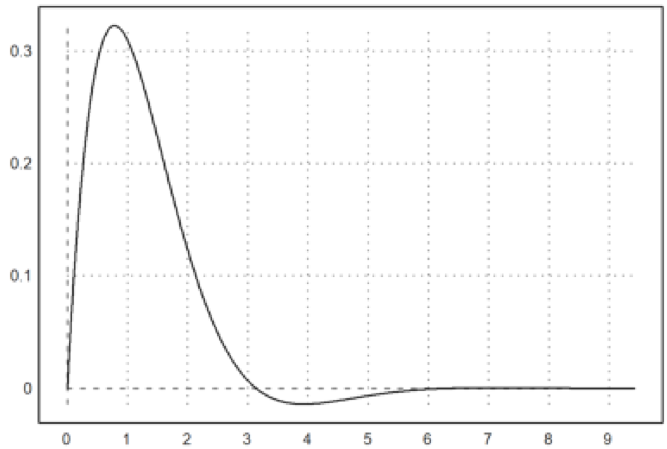


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.

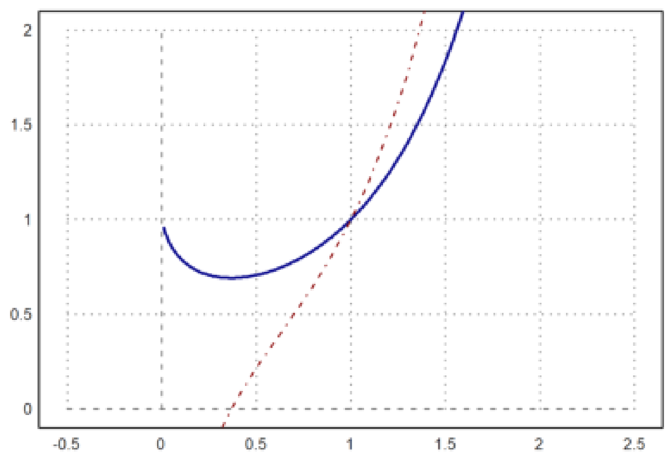
```
>expr &= sin(x)*exp(-x)
```

```
                    - x
                  E     sin(x)
```

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

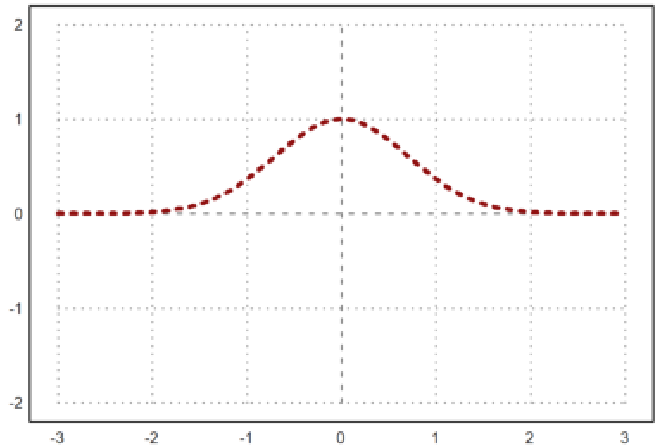



```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```



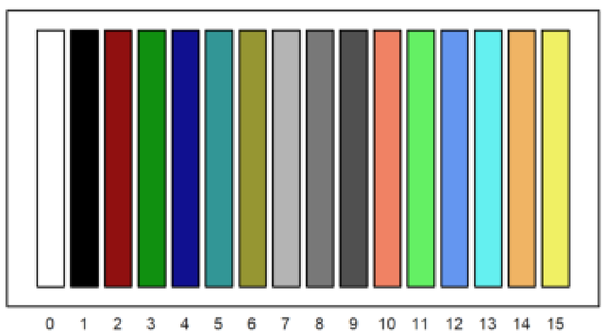
Untuk gaya garis ada berbagai pilihan.
- gaya="...". Pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- warna: Lihat di bawah untuk warna.
- ketebalan: Default adalah 1.
Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB.
- 0.15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye terang, kuning
- rgb(merah, hijau, biru): parameter adalah real dalam [0,1].

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



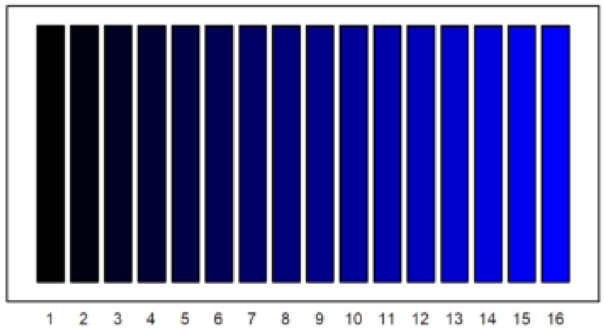
Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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



Tapi Anda bisa menggunakan warna apa saja.

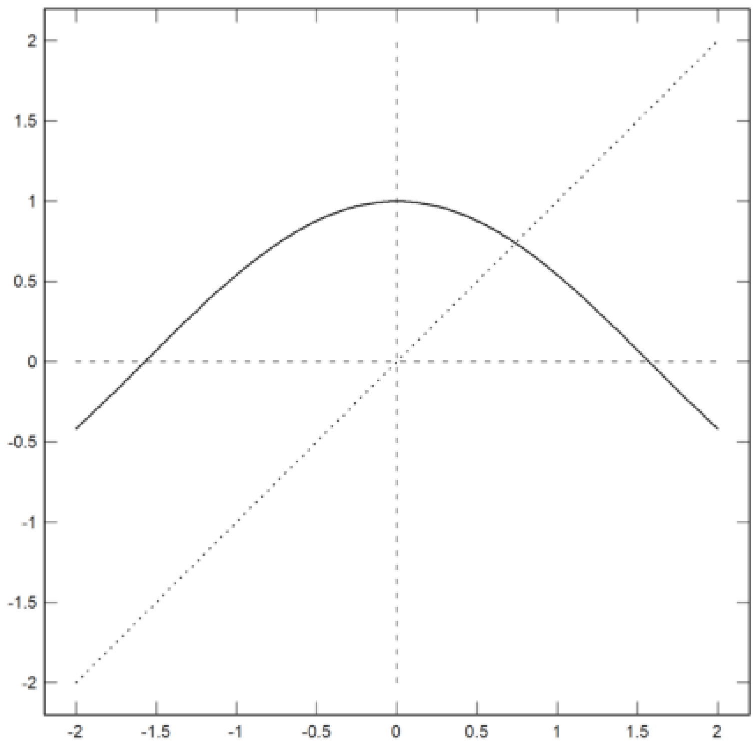
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



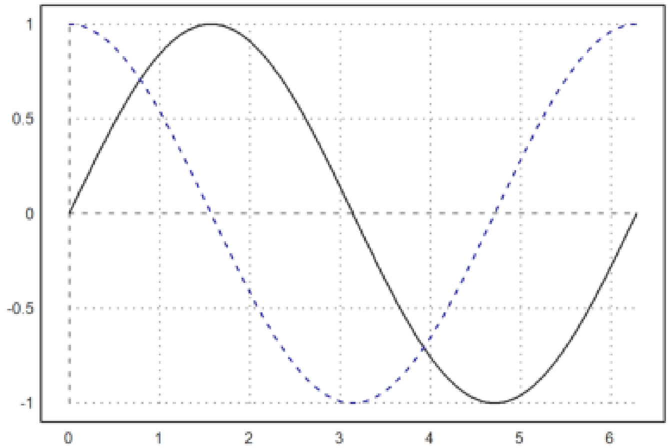
Menggambar Beberapa Kurva pada bidang koordinat yang sama

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

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

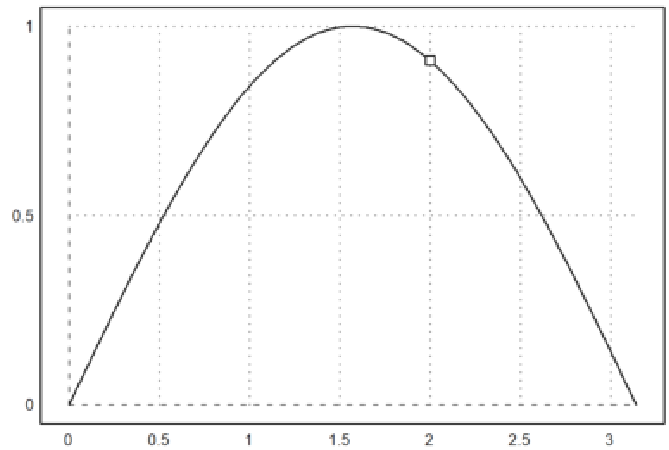


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



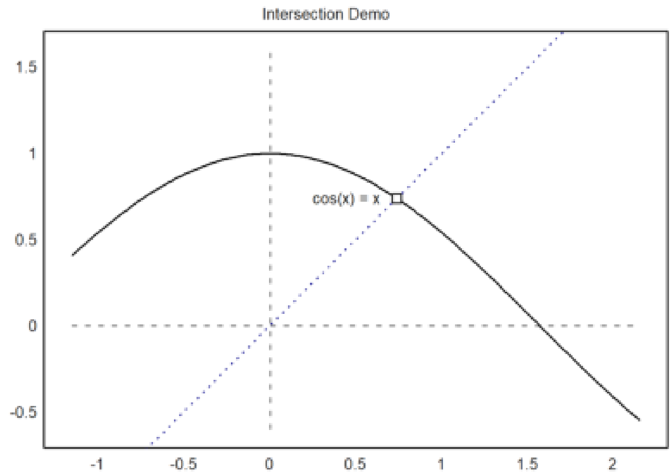
Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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



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

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

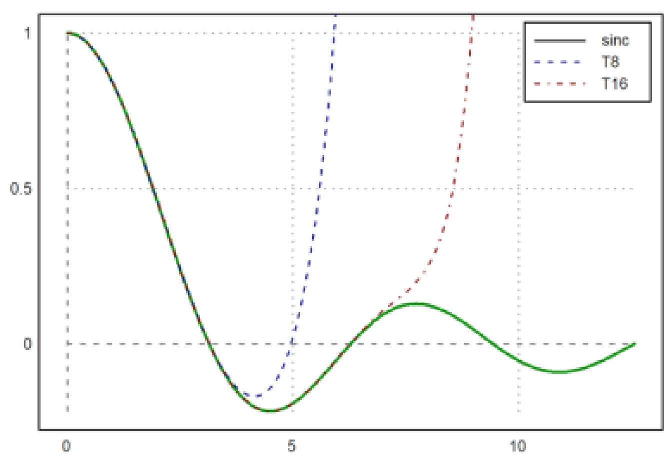


Dalam demo berikut, kami memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolis.
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke plot2d(). Yang kedua dan yang ketiga memiliki set flag >add, yang membuat plot menggunakan rentang sebelumnya.
Kami menambahkan kotak label yang menjelaskan fungsi.

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

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

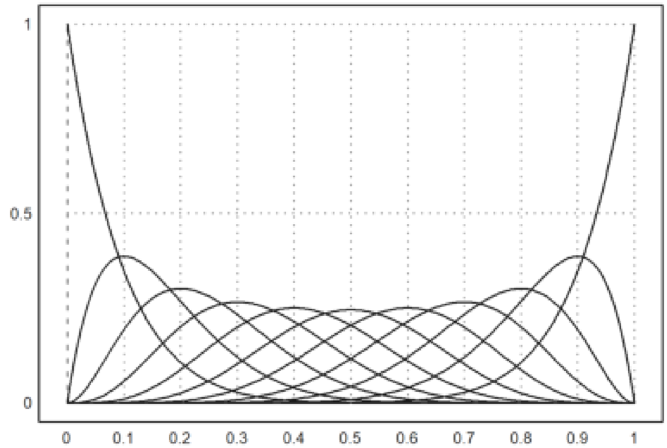
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```



Dalam contoh berikut, kami menghasilkan Bernstein-Polinomial.

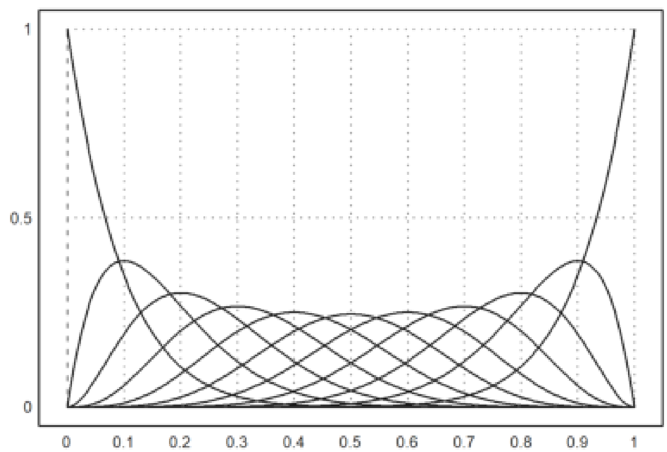
$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

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



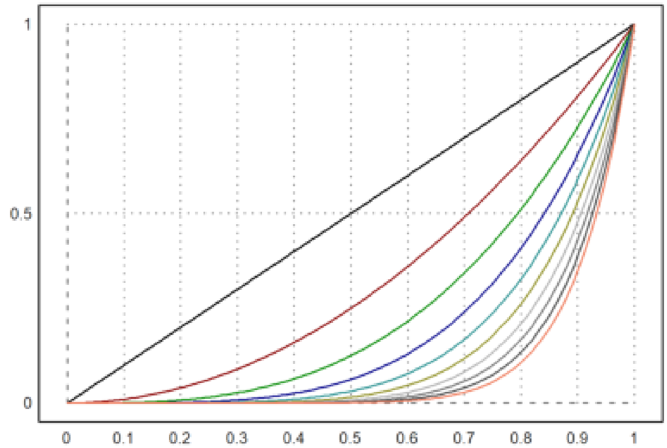
Metode kedua menggunakan pasangan matriks nilai-x dan matriks nilai-y yang berukuran sama.
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

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



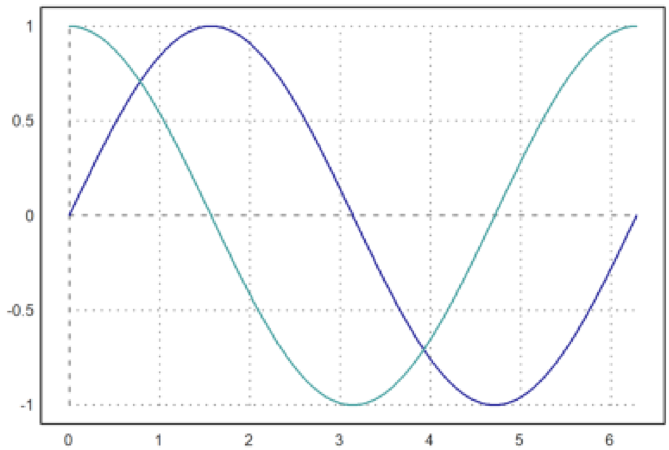
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

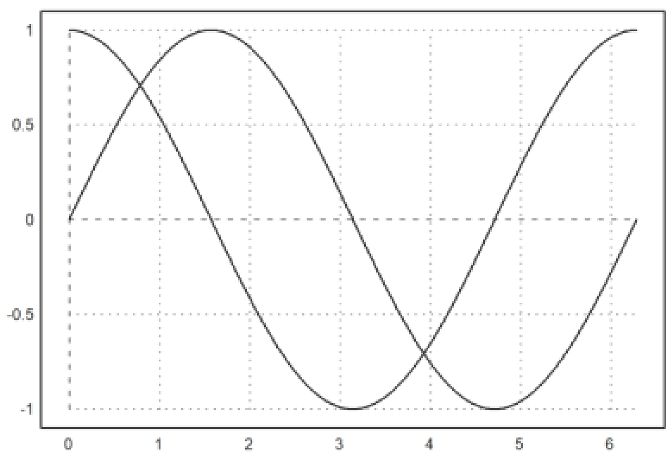


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

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




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



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

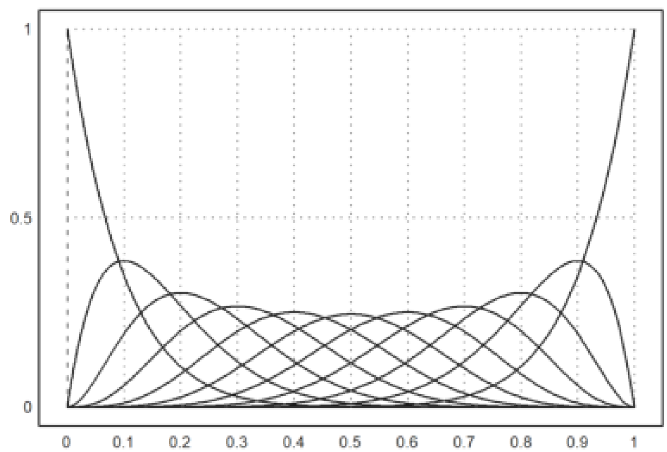
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

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

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

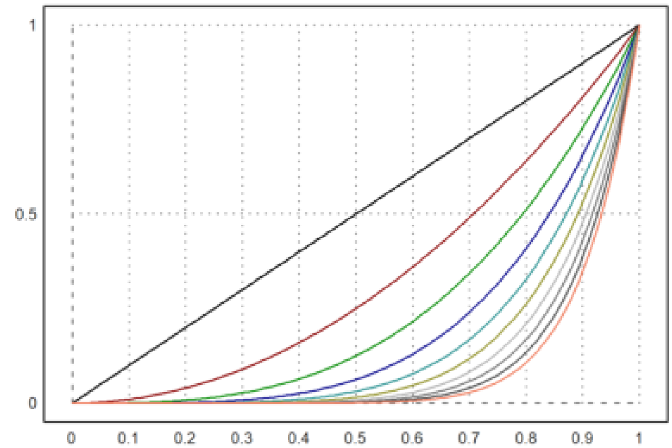
```
(1-x)^10
10*(1-x)^9*x
45*(1-x)^8*x^2
120*(1-x)^7*x^3
210*(1-x)^6*x^4
252*(1-x)^5*x^5
210*(1-x)^4*x^6
120*(1-x)^3*x^7
45*(1-x)^2*x^8
10*(1-x)*x^9
x^10
```

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



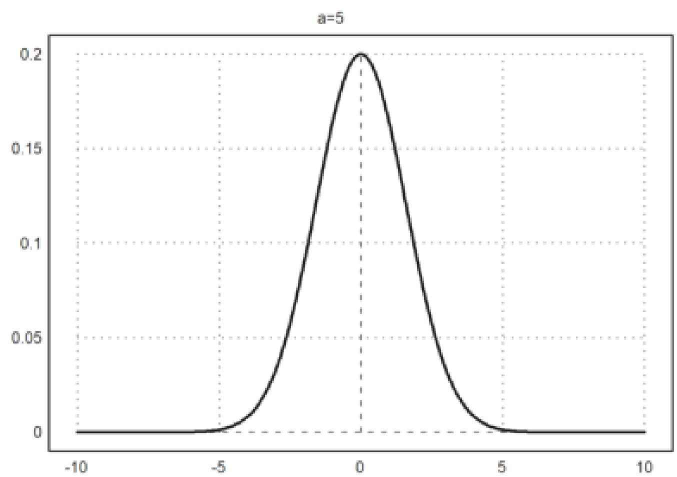
Alternatif lain adalah dengan menggunakan bahasa matriks Euler.
Jika ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan diplot ke dalam satu plot.
Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika array warna ditambahkan, itu akan digunakan untuk setiap baris plot.

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



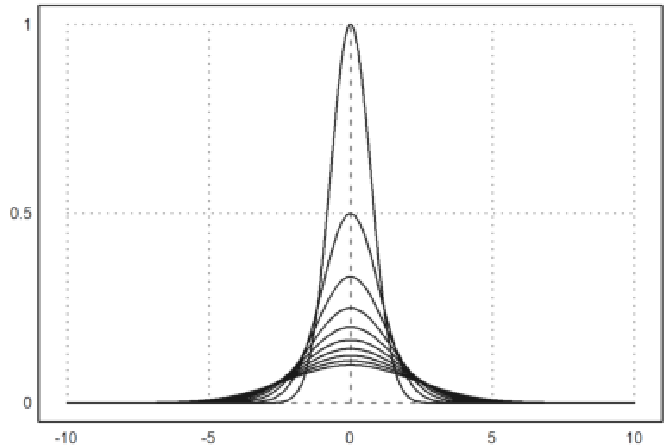
Ekspresi dan fungsi satu baris dapat melihat variabel global.
Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.
Berhati-hatilah, untuk meletakkan semua parameter yang ditetapkan di akhir perintah plot2d. Dalam contoh kita meneruskan a=5 ke fungsi f, yang kita plot dari -10 hingga 10.

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



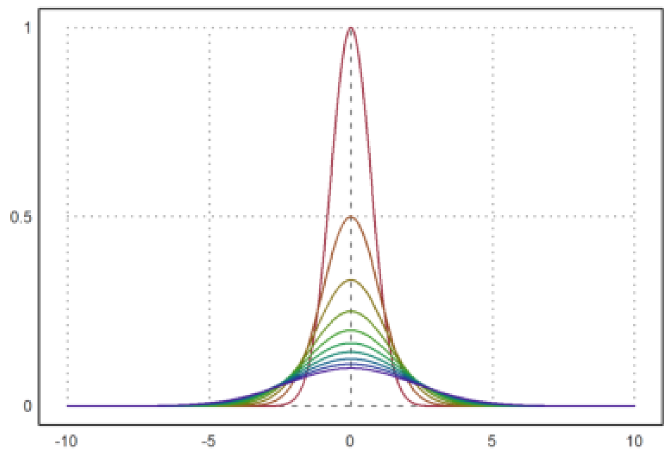
Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk meneruskan argumen ke fungsi yang dengan sendirinya diteruskan sebagai argumen ke fungsi lain.
Dalam contoh berikut, kami menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

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



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

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

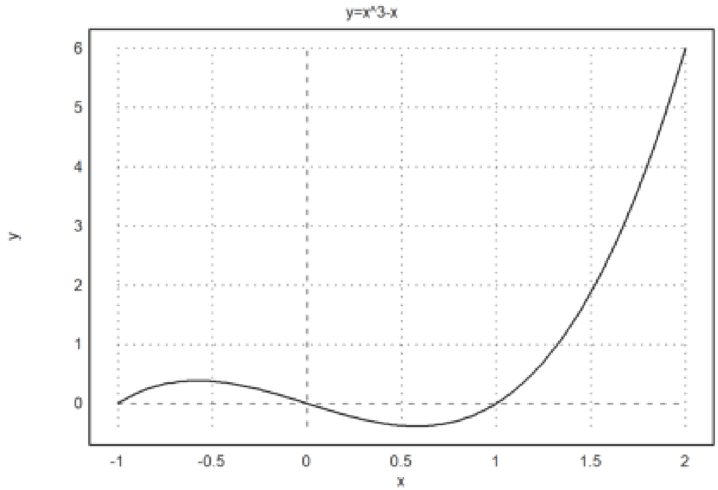


Label Teks

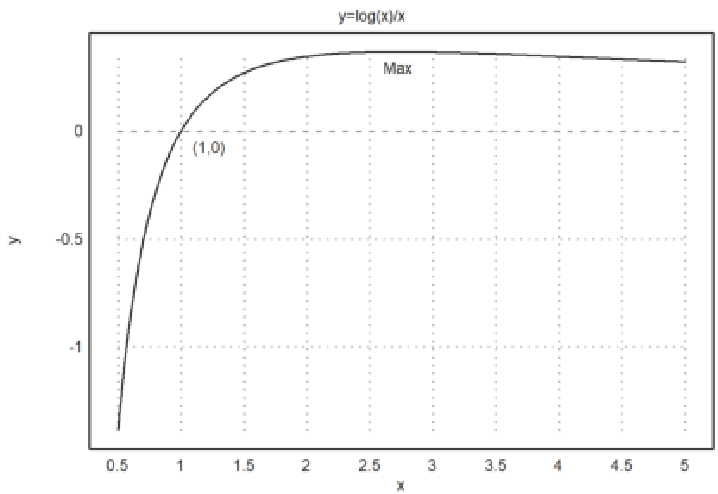
Dekorasi sederhana bisa
- judul dengan judul="..."
- x- dan y-label dengan xl="...", yl="..."
- label teks lain dengan label("...",x,y)

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"):
```

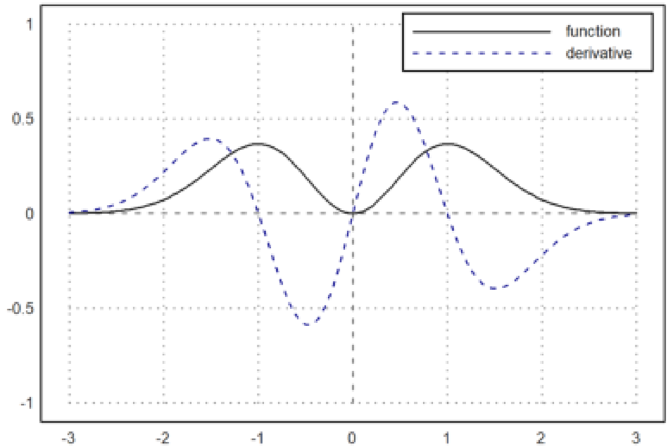



```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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):
```

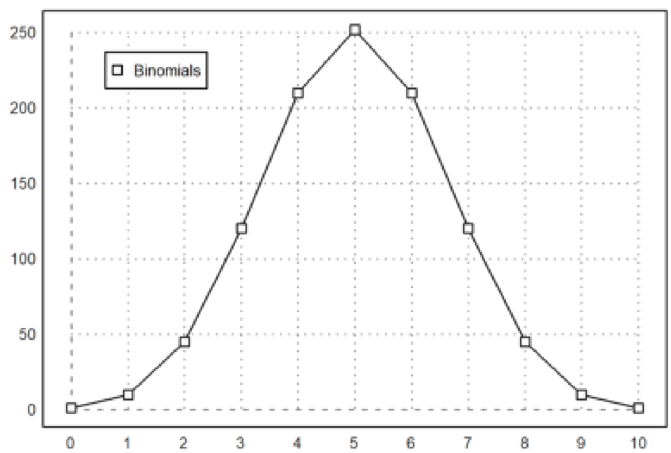


Kotak ditambatkan di kanan atas secara default, tetapi > kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.

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

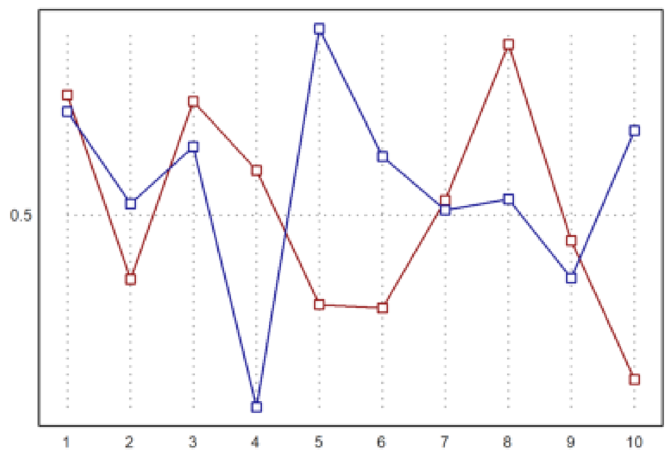
Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

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



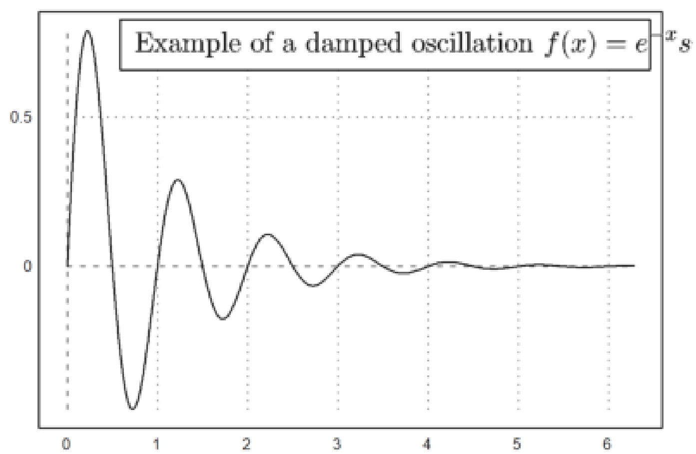
Gaya plot ini juga tersedia di statplot(). Seperti di plot2d() warna dapat diatur untuk setiap baris plot. Ada lebih banyak plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

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



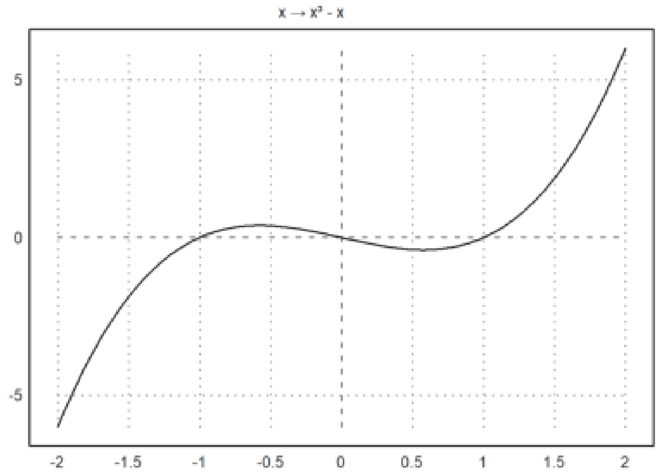
Fitur serupa adalah fungsi textbox().
Lebar secara default adalah lebar maksimal dari baris teks. Tapi itu bisa diatur oleh pengguna juga.

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



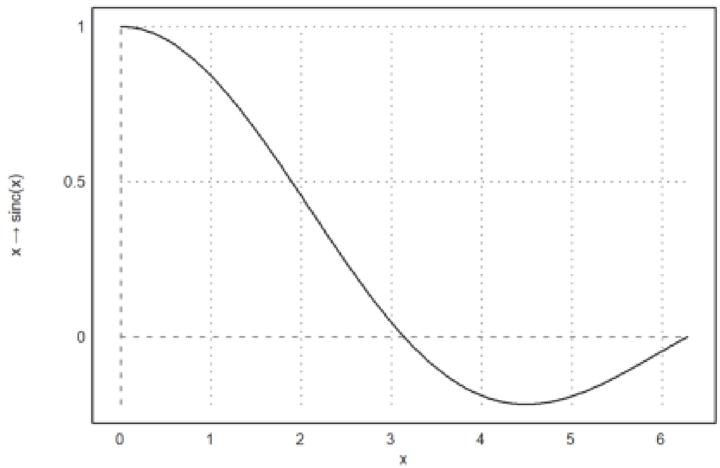
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

```
>plot2d("x^3-x",title=u"x &rarr; x&sup3; - 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):
```



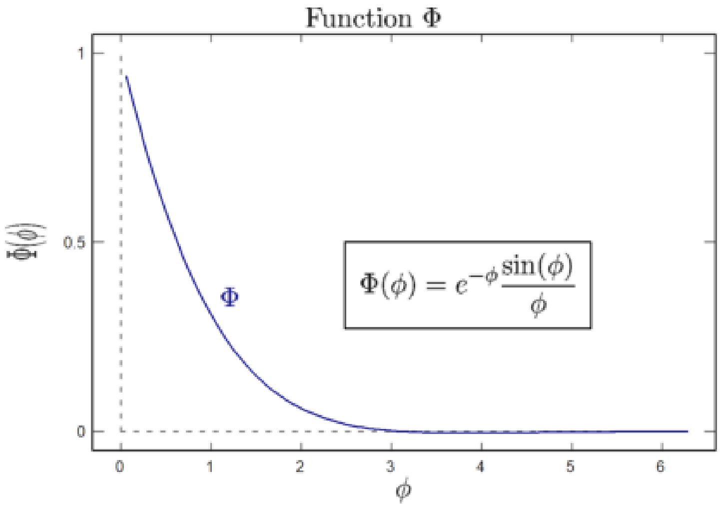
LaTeX

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).

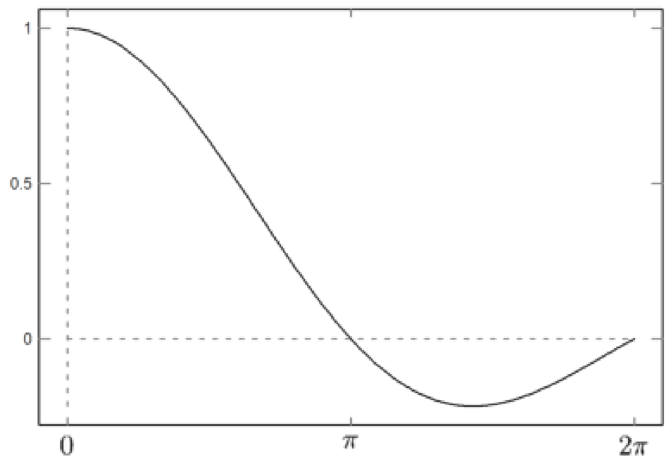
Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

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



Seringkali, kami menginginkan spasi dan label teks non-konformal pada sumbu x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.
Cara termudah adalah dengan membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Dalam contoh berikut, kami menggunakan tiga string LaTeX untuk label pada sumbu x dengan xtick().

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



Tentu saja, fungsi juga dapat digunakan.

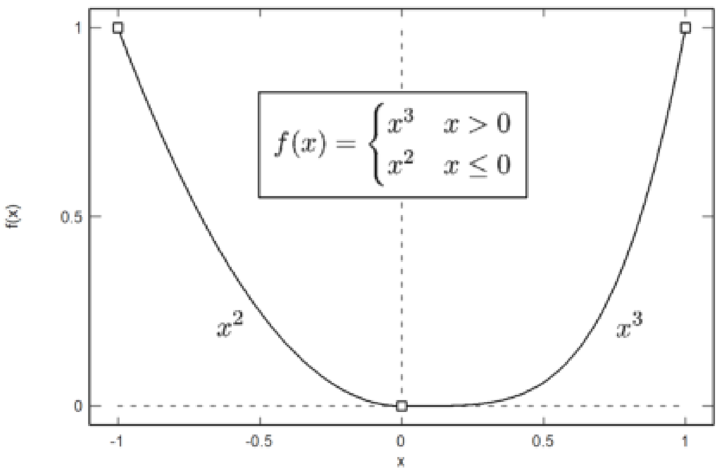
```
>function map f(x) ...
```

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

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

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

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

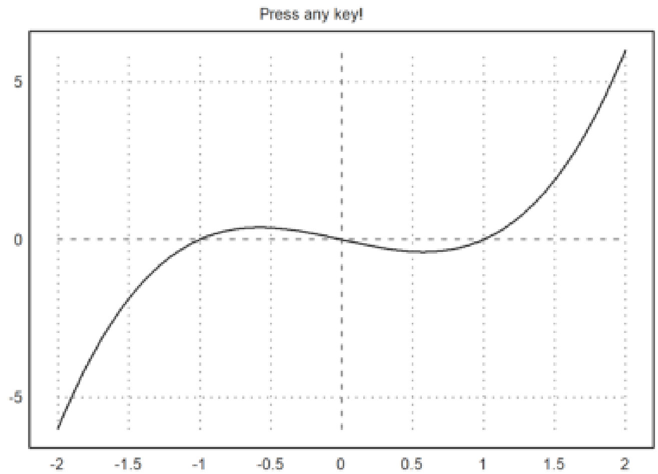


Interaksi pengguna

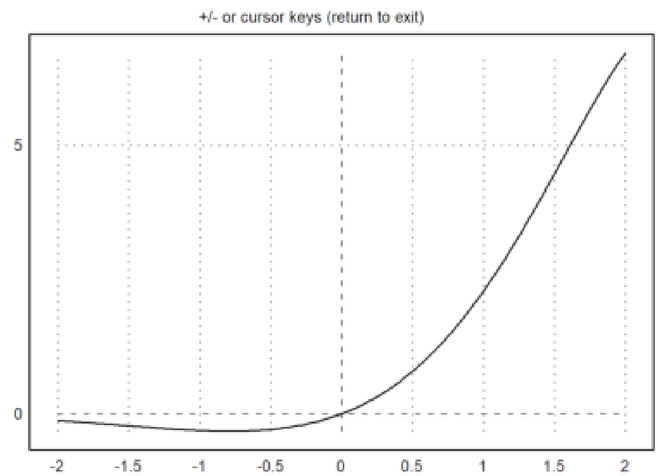
Saat memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat
- perbesar dengan + atau -
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali

Tombol spasi akan mengatur ulang plot ke jendela plot asli.
Saat memplot data, flag >user hanya akan menunggu penekanan tombol.

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




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



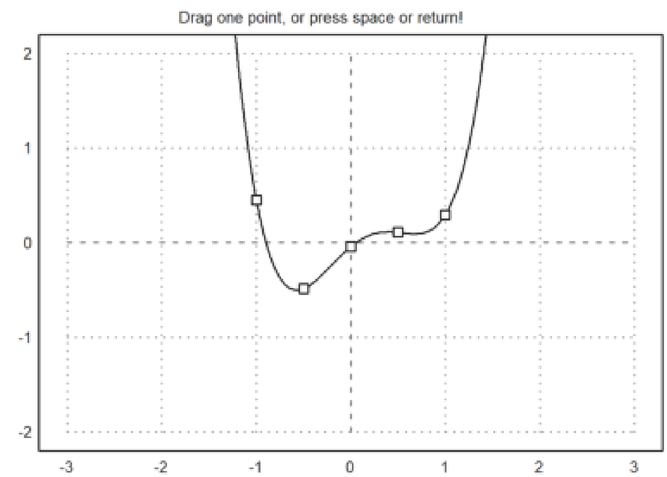
Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).
Fungsi bawaan mousedrag() menunggu event mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse dipindahkan atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.
Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

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

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

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.
Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

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

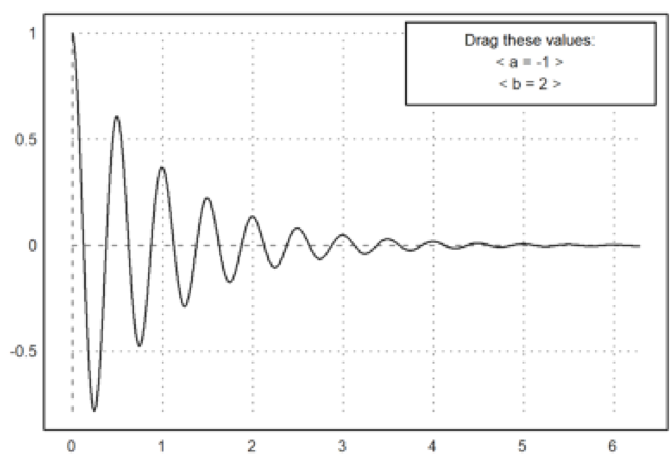


Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.
Pertama kita membutuhkan fungsi plot.

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

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.
Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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



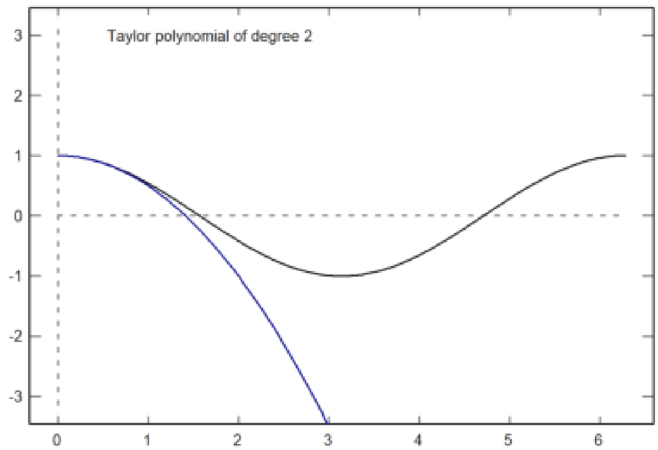
Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

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

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

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

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



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

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

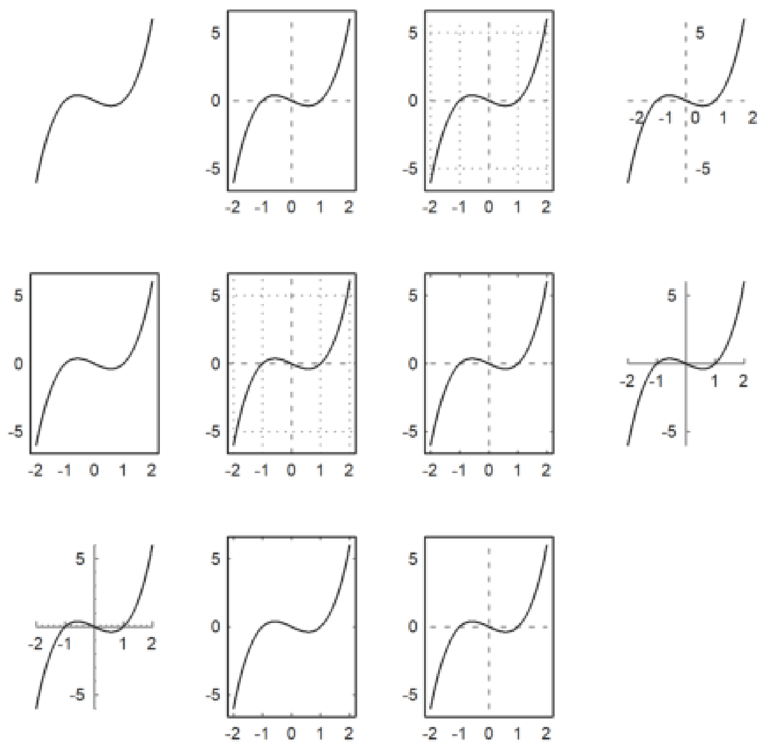
```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");
    start=0;
    repeat
      {flag,m,time}=mousedrag();
      if flag==0 then return; endif;
      if flag==2 then
        hold on; mark(m[1],m[2]); hold off;
      endif;
    end
  endfunction
```

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

Gaya Plot 2D

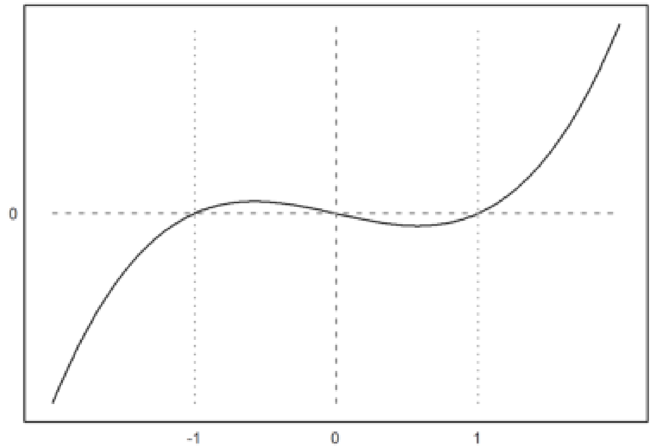
Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

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



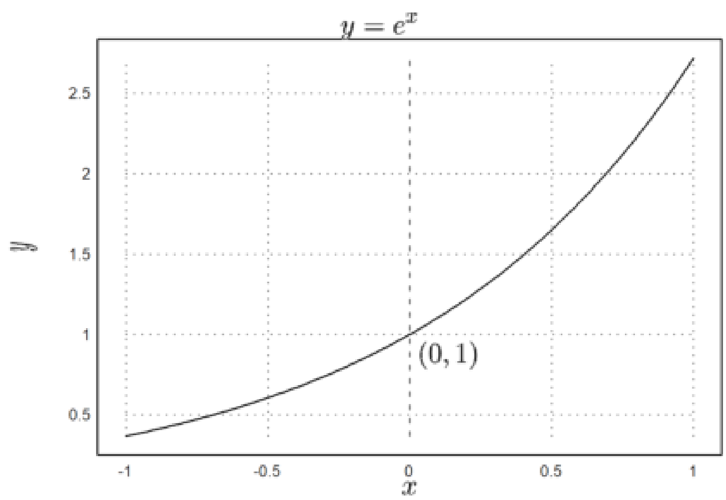
Parameter <frame mematikan frame, dan framecolor=blue mengatur frame ke warna biru.
Jika Anda ingin centang sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

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



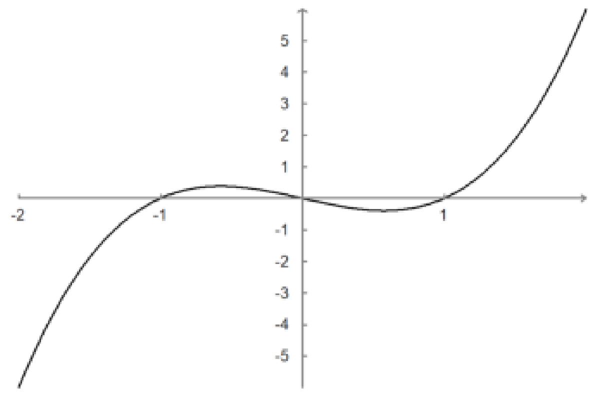
Untuk judul plot dan label sumbu, lihat contoh berikut.

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



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

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

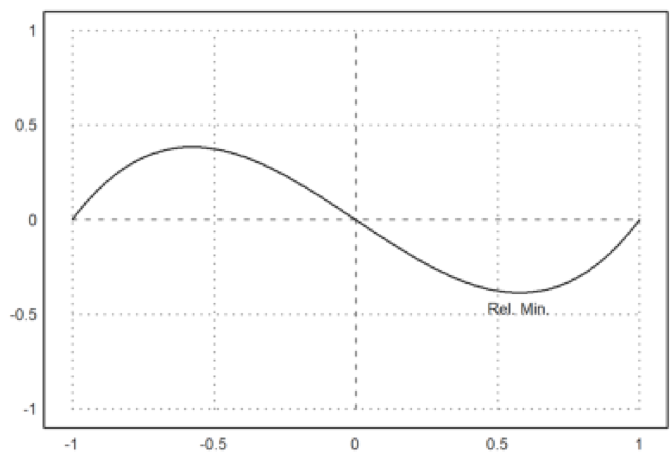


Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

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

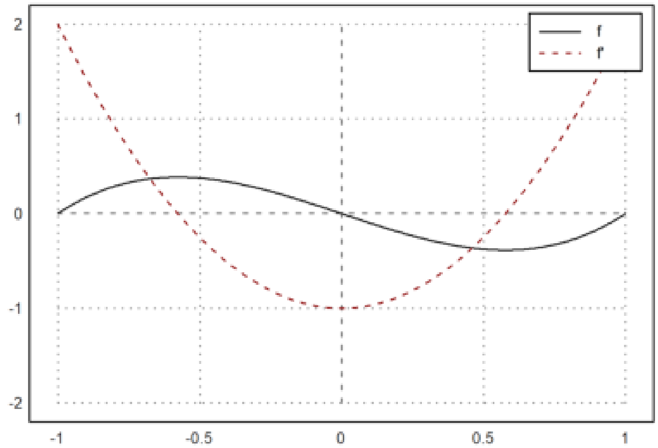
```
                                 3
                                x  - x
```

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

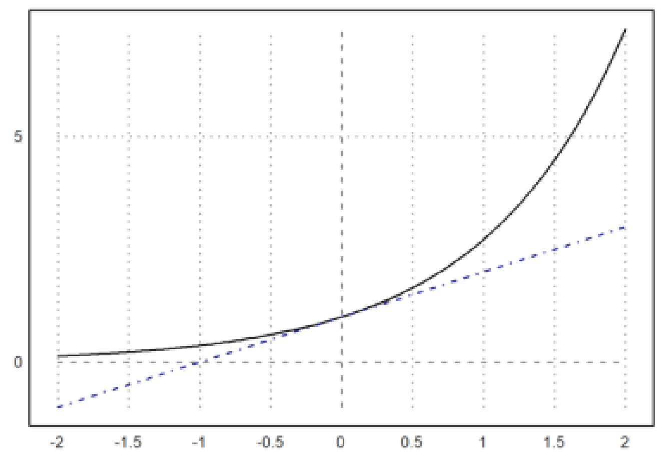


Ada juga kotak teks.

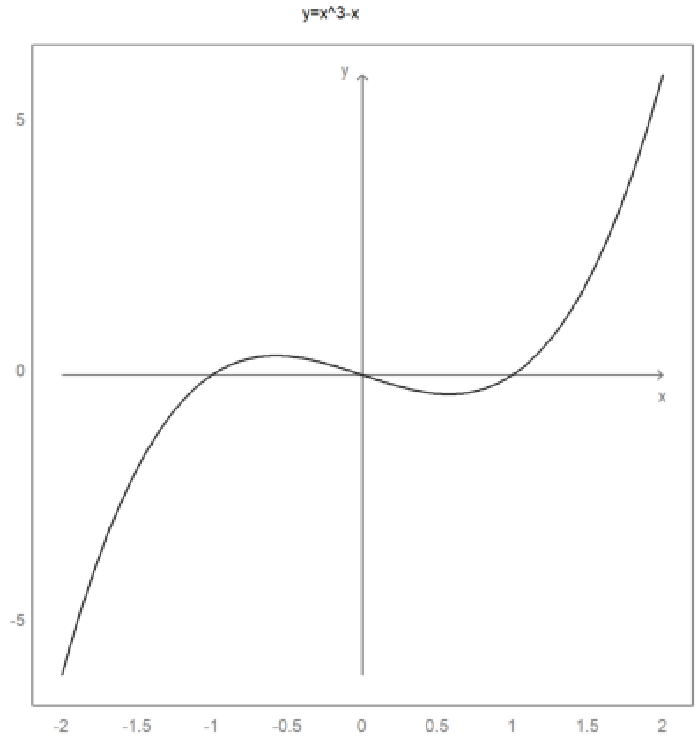
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```




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

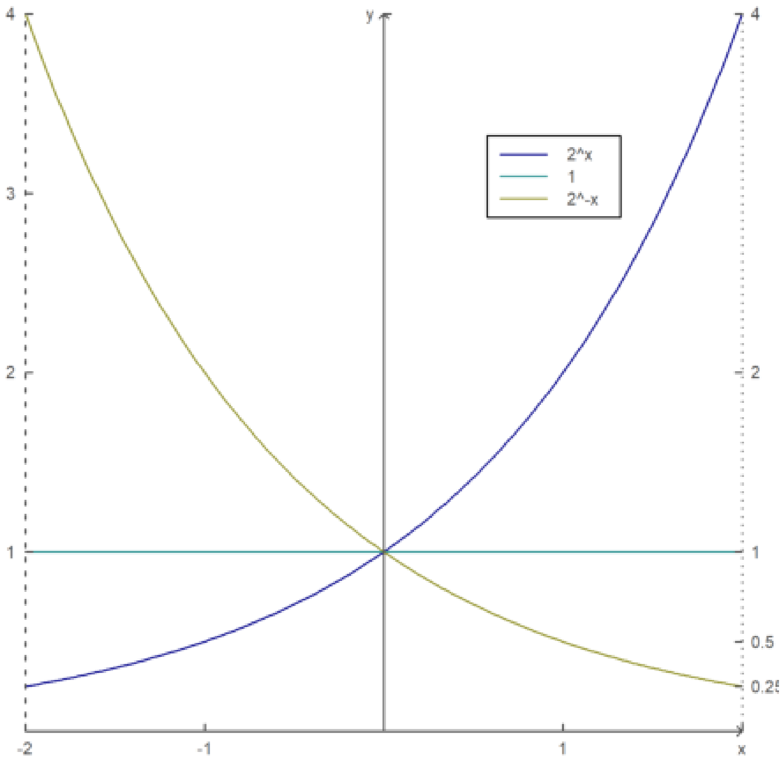


```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



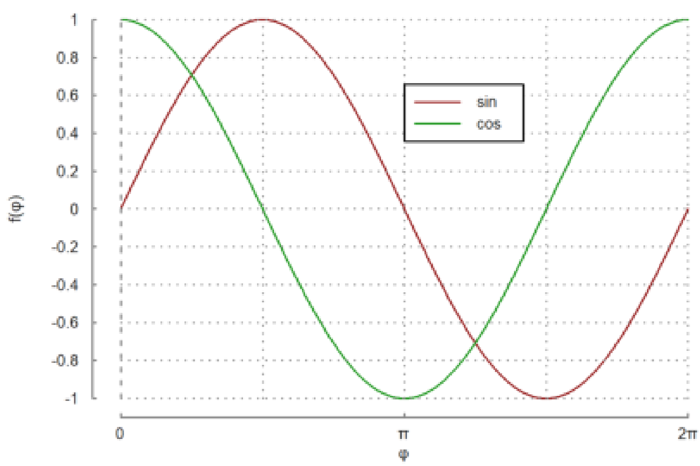
Untuk kontrol lebih, sumbu x dan sumbu y dapat dilakukan secara manual.
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

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



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

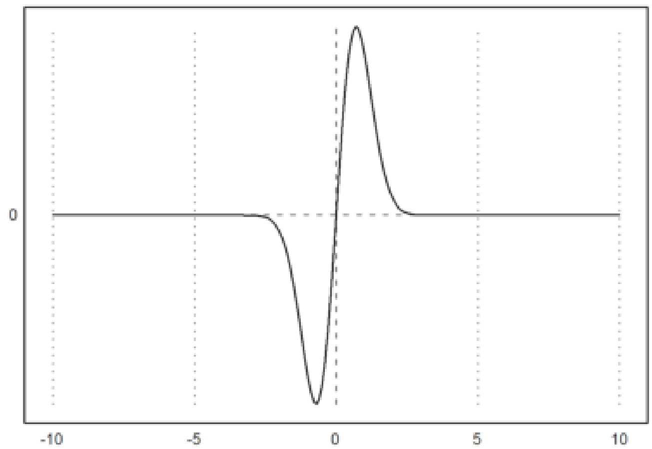
```
>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0",u"&pi;",u"2&pi;"],style="-",>ticks,>zero);  ...
> xgrid((0:0.5:2)*pi,<ticks); ...
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left);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



Merencanakan Data 2D

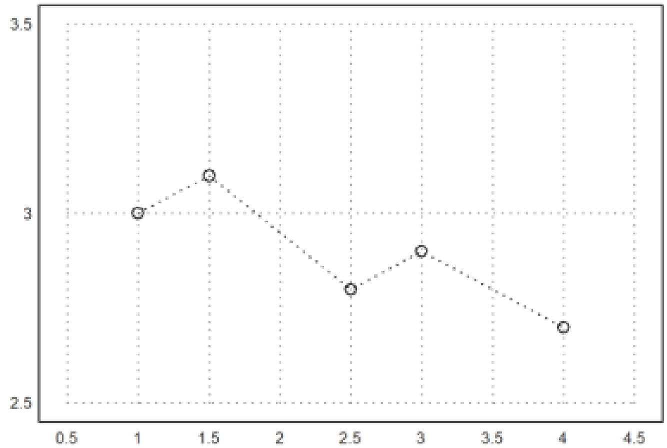
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari suatu kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, >persegi dapat diatur untuk menjaga rasio aspek persegi.
Memplot ekspresi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

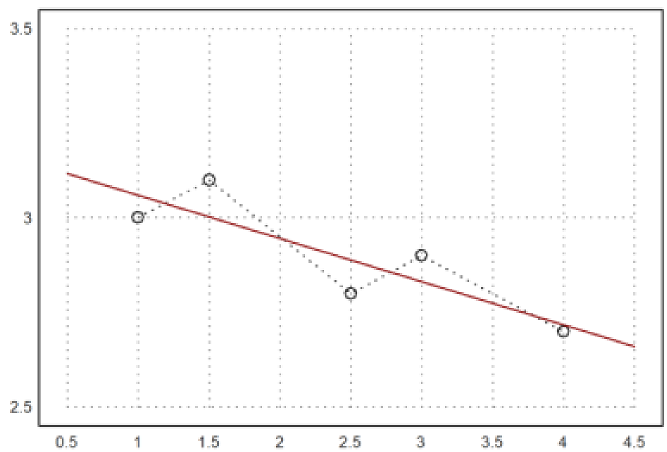


Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.
- style="...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "< >", "o", "..", "", "|".
Untuk memplot set poin gunakan >points. Jika warna adalah vektor 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
```



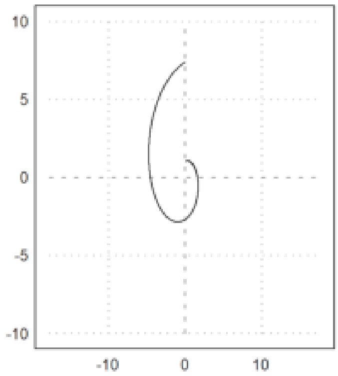

```
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red): // add plot of line
```

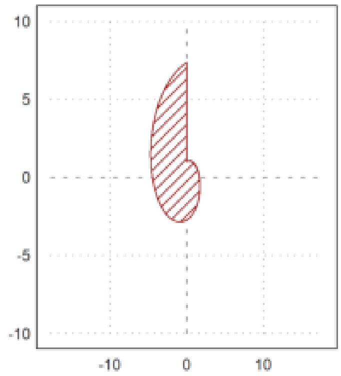


Menggambar Daerah Yang Dibatasi Kurva

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.

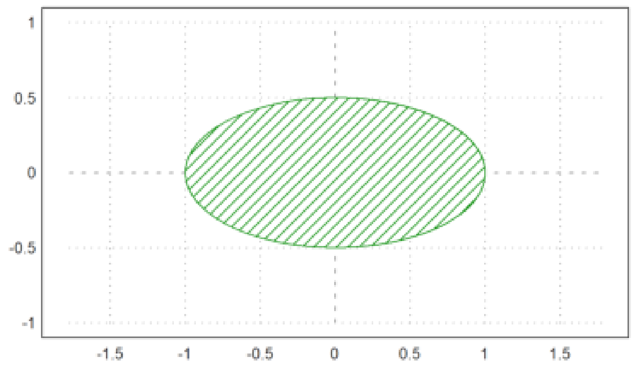
```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```



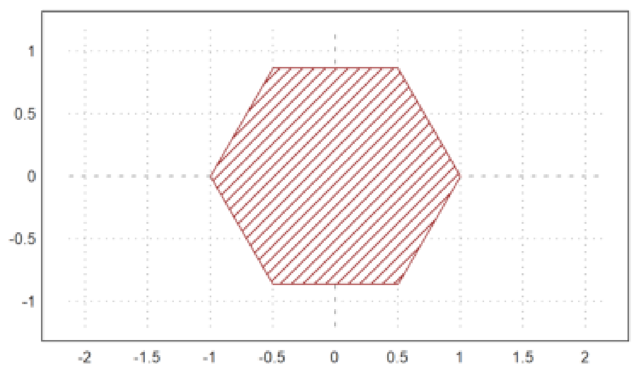


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

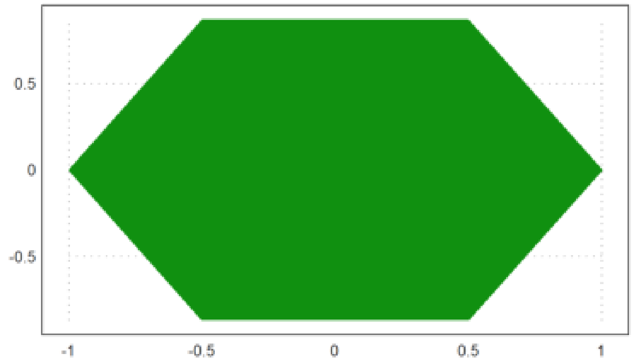
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




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

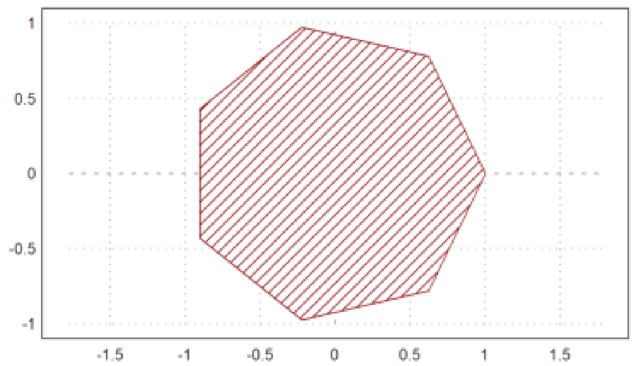



```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



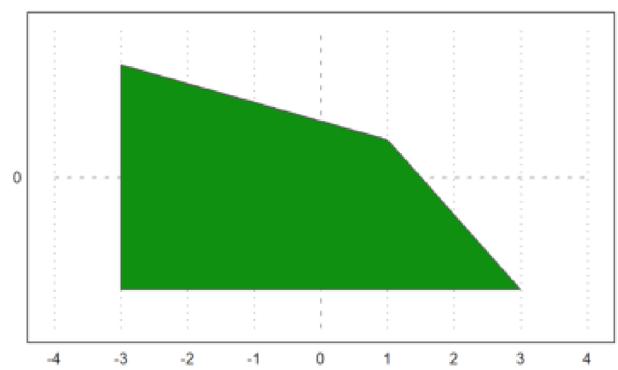
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>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<=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):
```

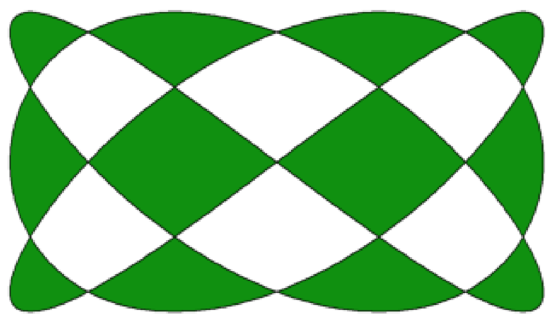


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

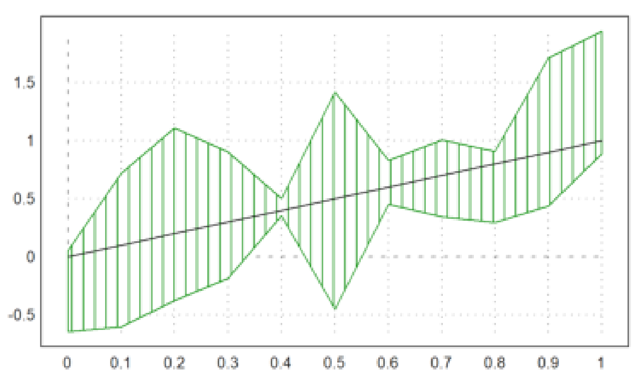
Kami sekarang memiliki vektor x dan y nilai. plot2d() dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk isi.

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



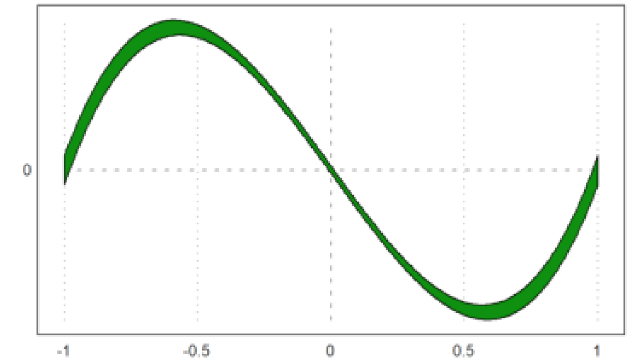
Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

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



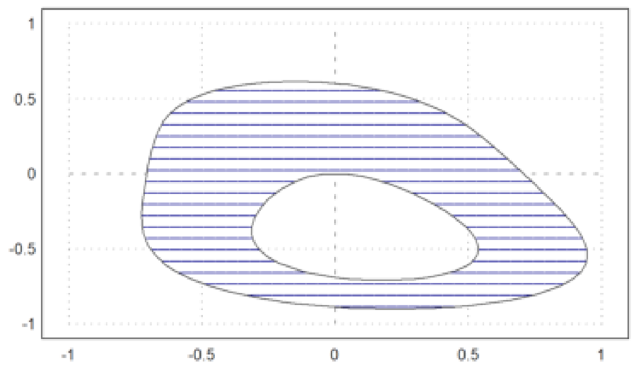
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>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.

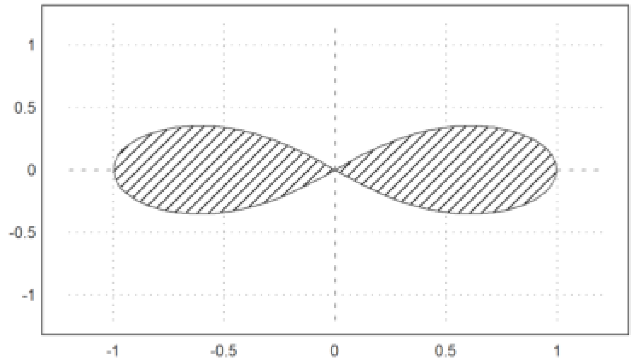
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



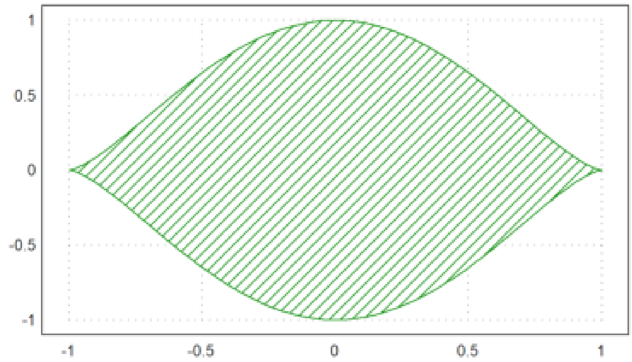
Kami juga dapat mengisi rentang nilai seperti

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

```
>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="/"):
```



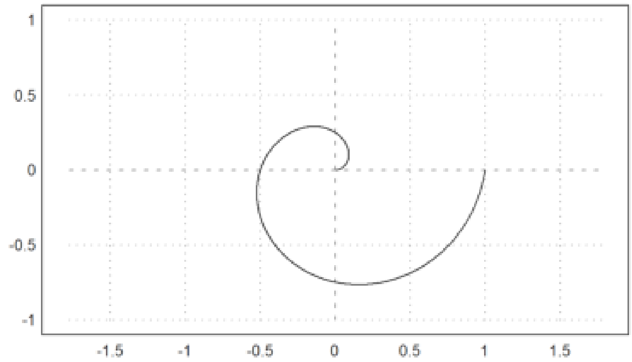
Grafik Fungsi Parametrik

Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.
Dalam contoh berikut, kami memplot spiral

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

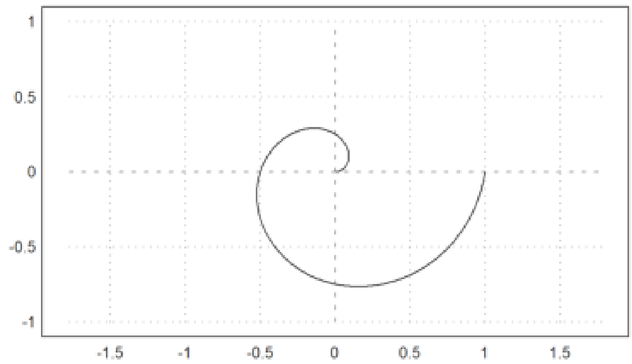
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

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

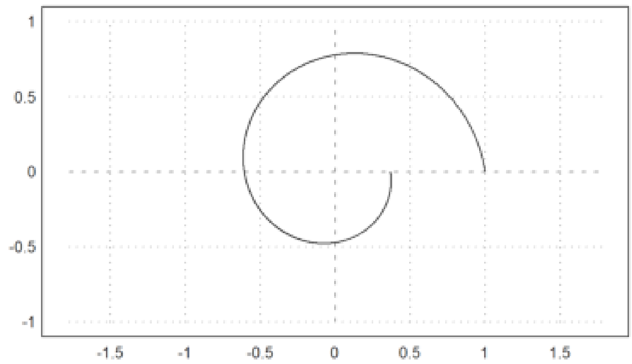


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
>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):
```



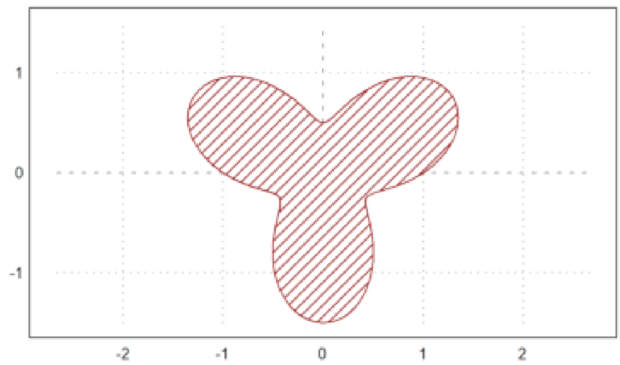
Dalam contoh berikutnya, kami memplot kurva

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

dengan

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

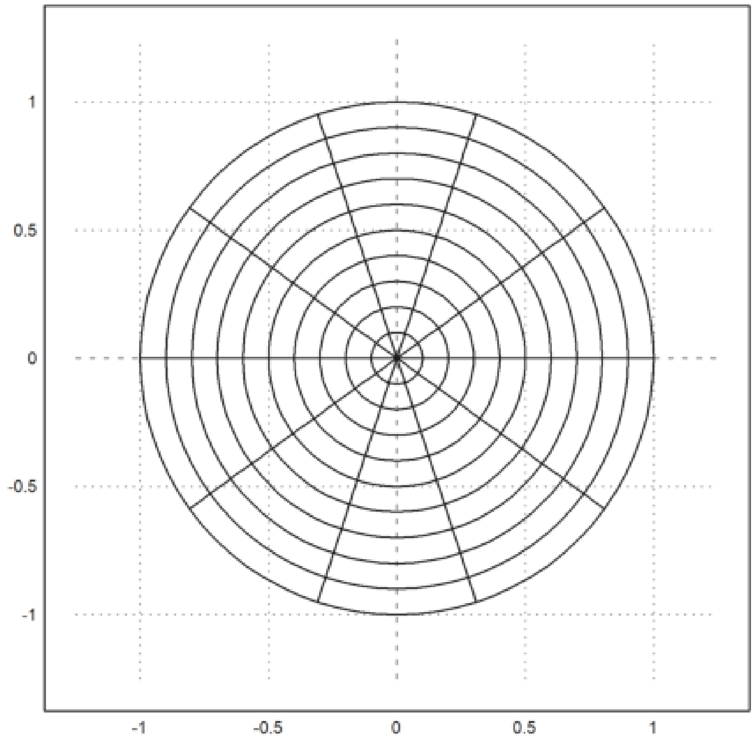
```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



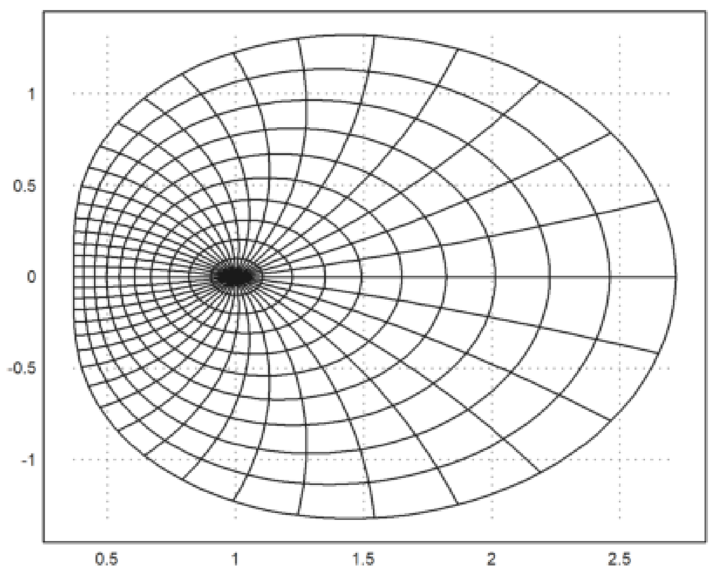
Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.
Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

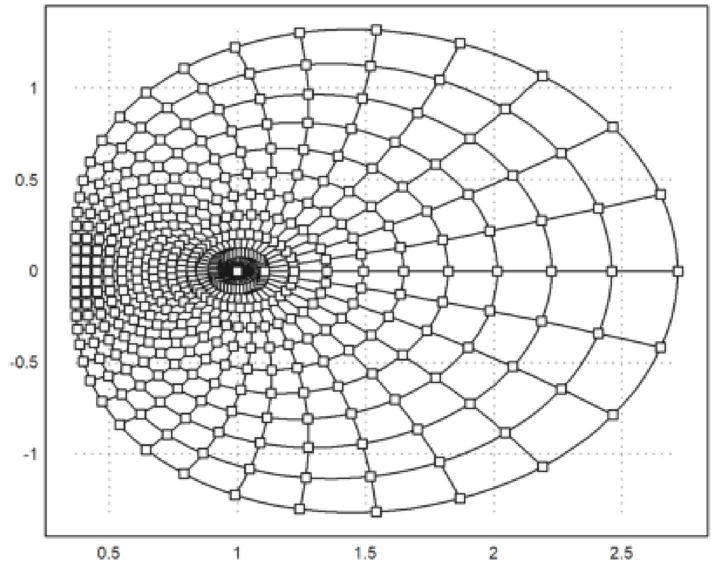
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




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



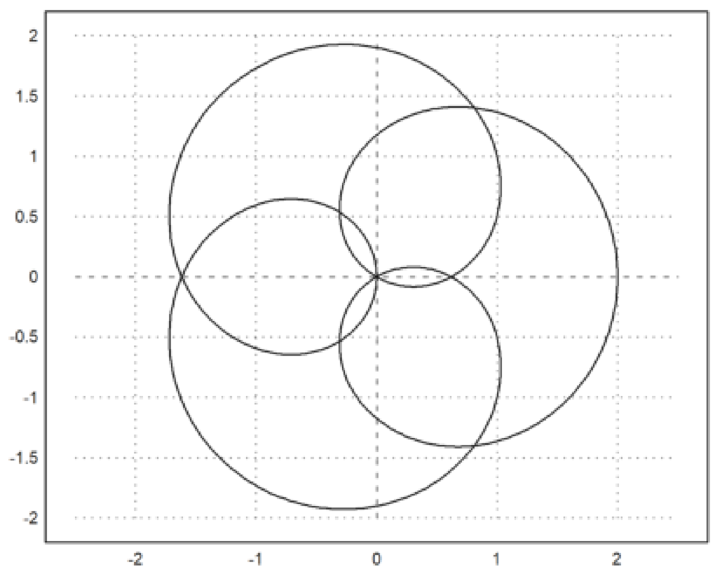

```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.
Dalam contoh, kami memplot lingkaran satuan dengan

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

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



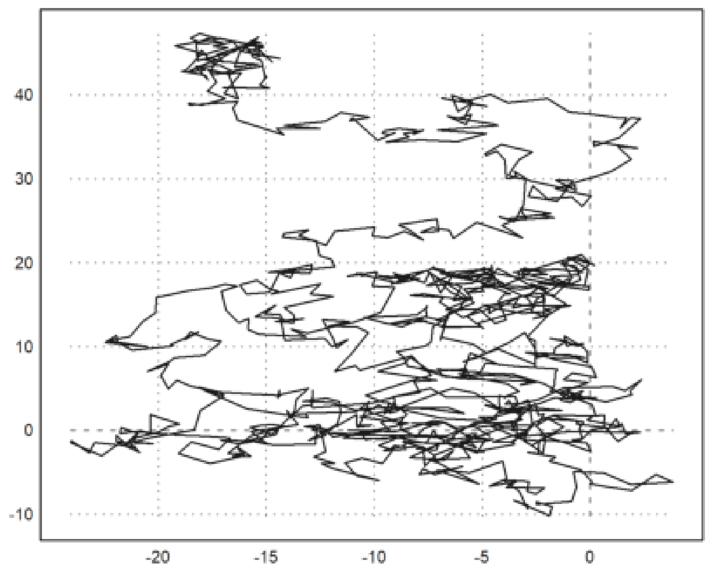
Plot Statistik

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.

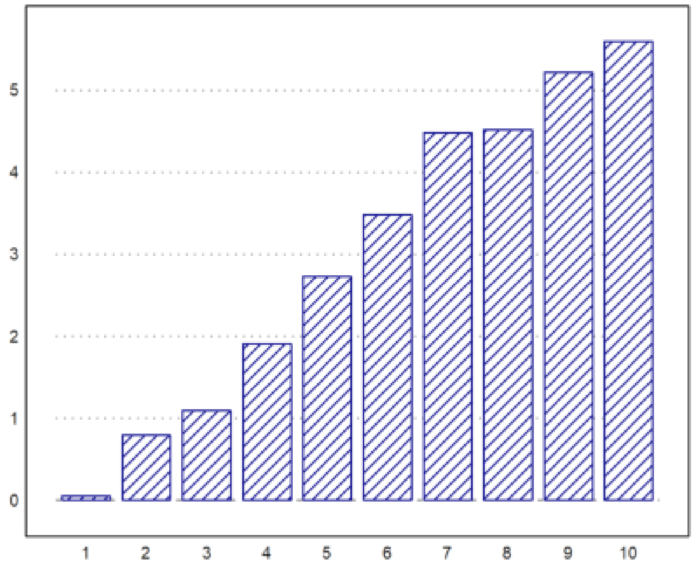
```
>plot2d(cumsum(randnormal(1,1000))):
```

Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

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

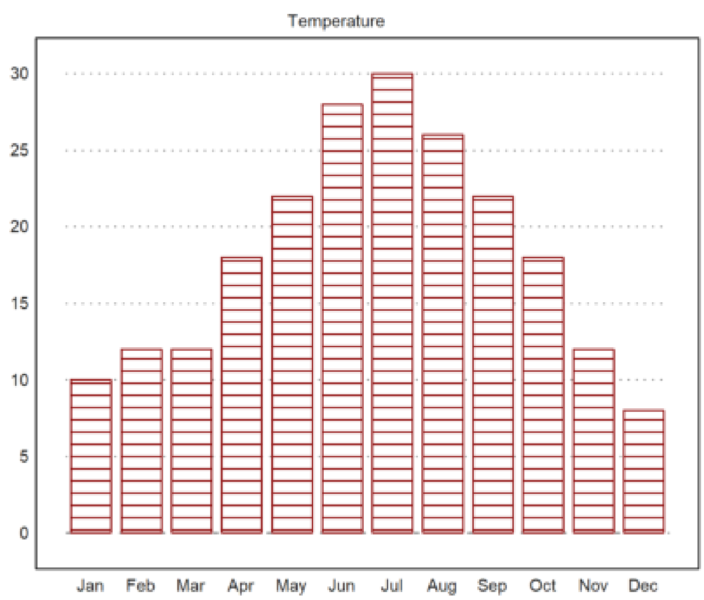



```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

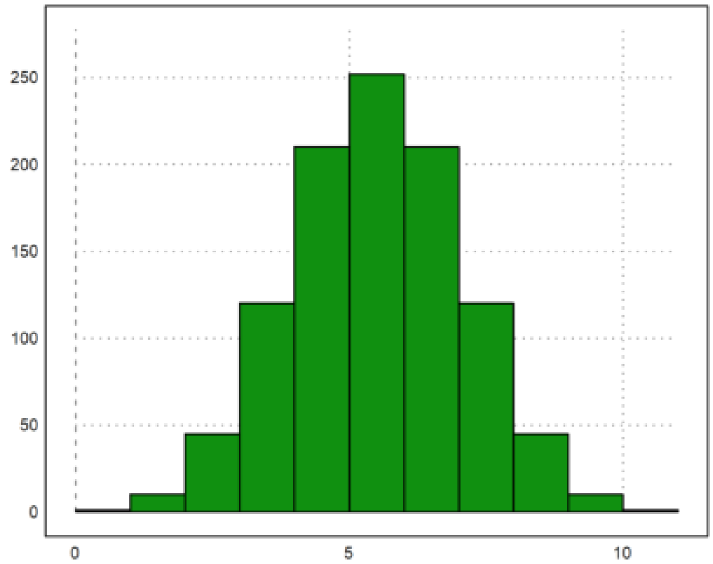


Itu juga dapat menampilkan string sebagai label.

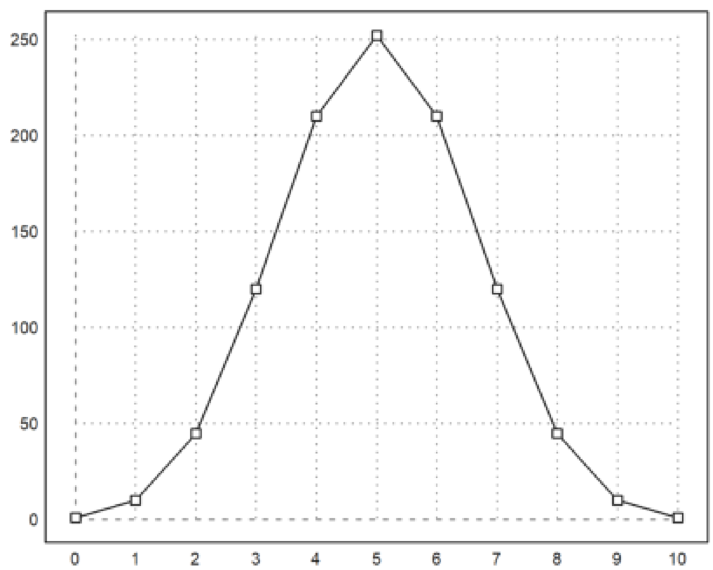
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



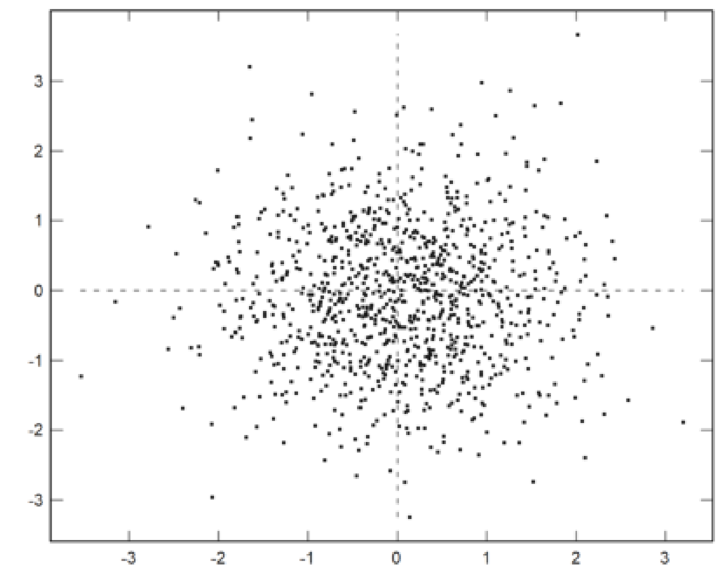

```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



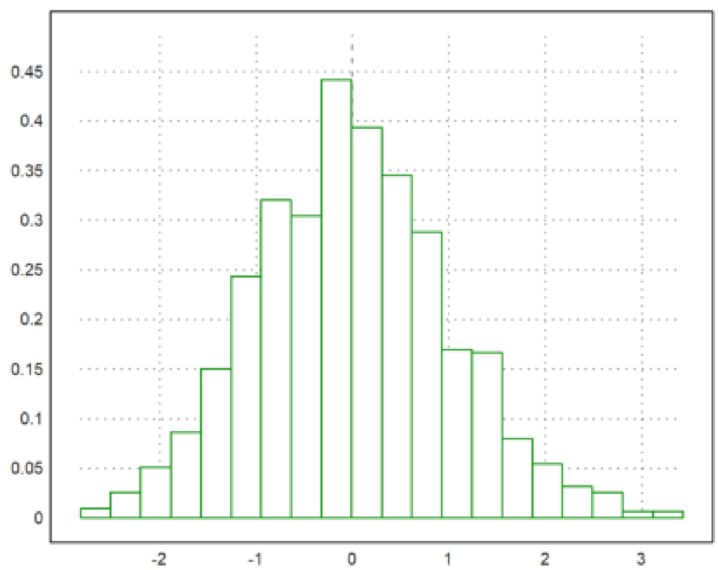

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



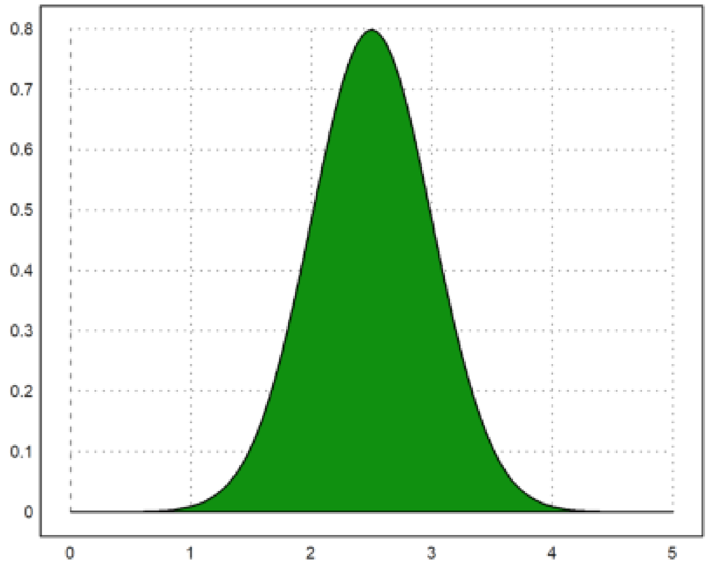

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```




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

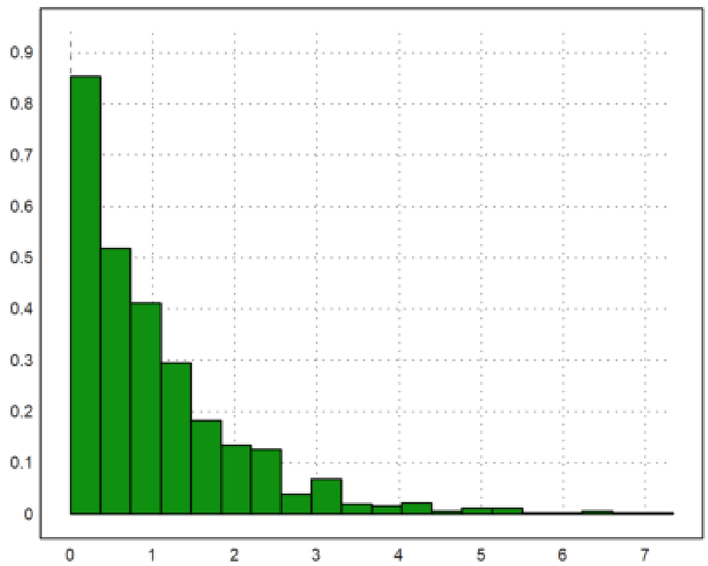



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



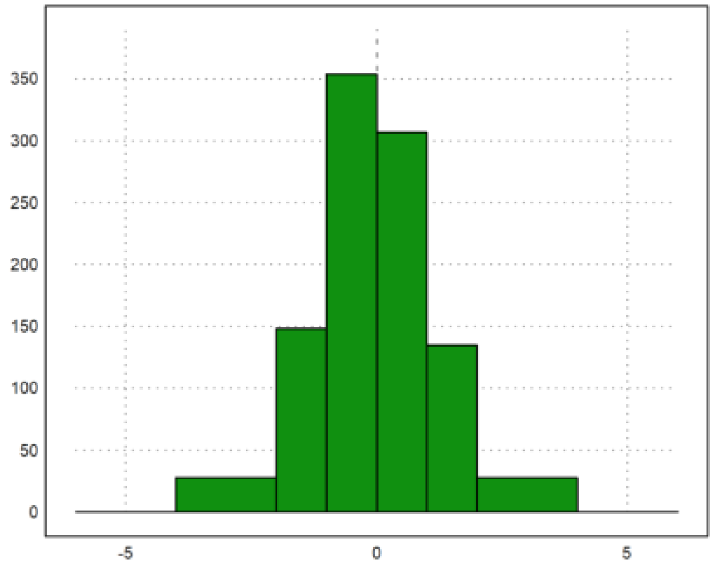
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



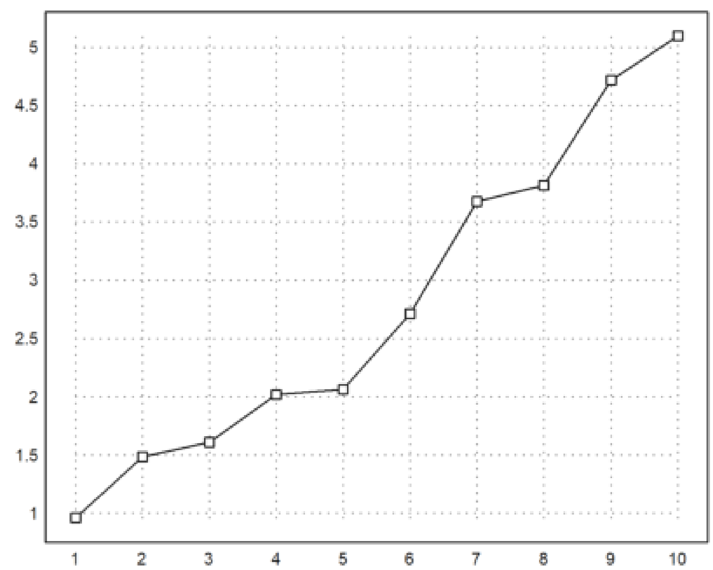
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>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):
```

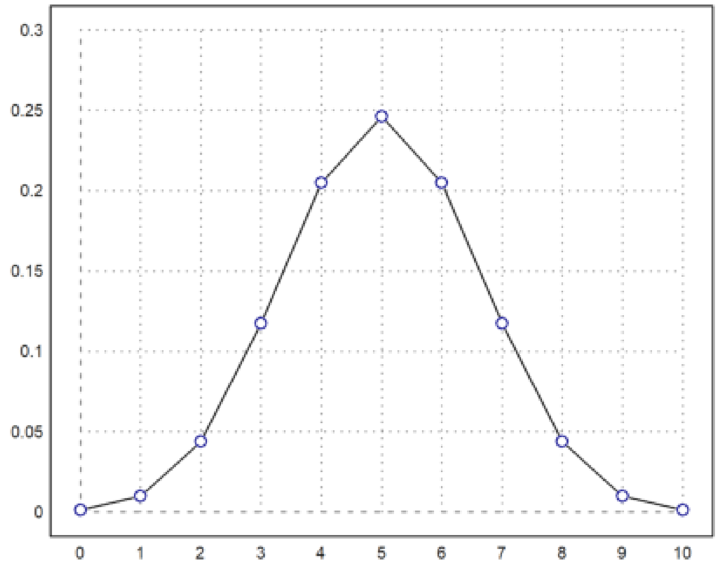


Fungsi statplot() menyetel gaya dengan string sederhana.

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

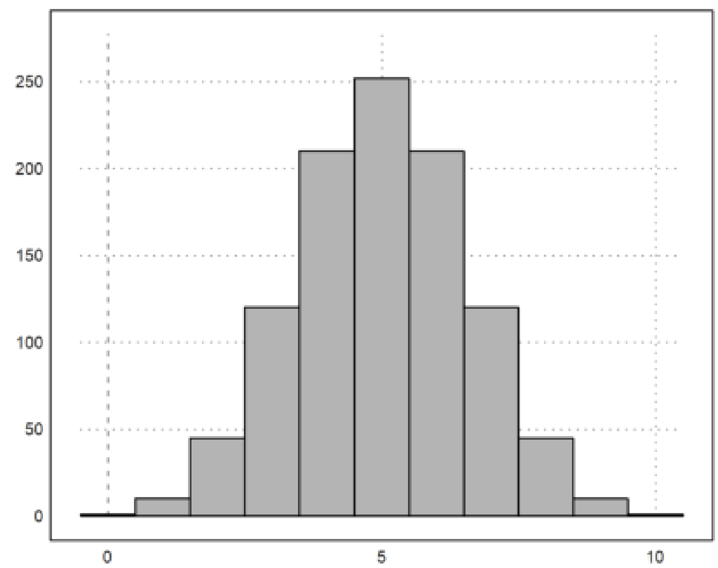


```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



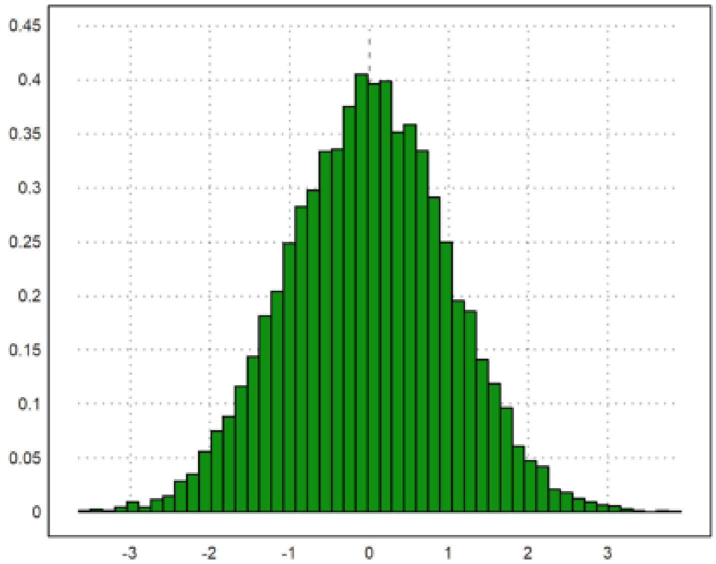
Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Bilah akan memanjang dari x[i] ke x[i+1] dengan nilai y[i]. Jika x memiliki ukuran yang sama dengan y, maka akan diperpanjang satu elemen dengan spasi terakhir.
Gaya isian dapat digunakan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

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

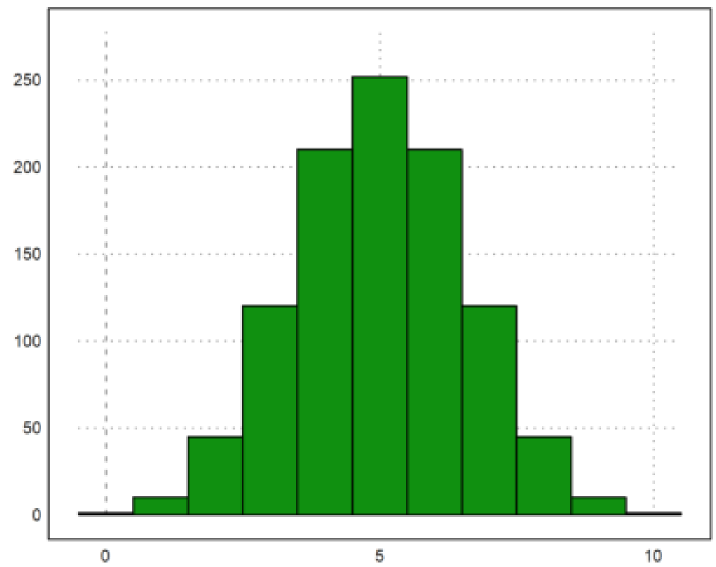


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat dinyatakan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >genap ditentukan, nilai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

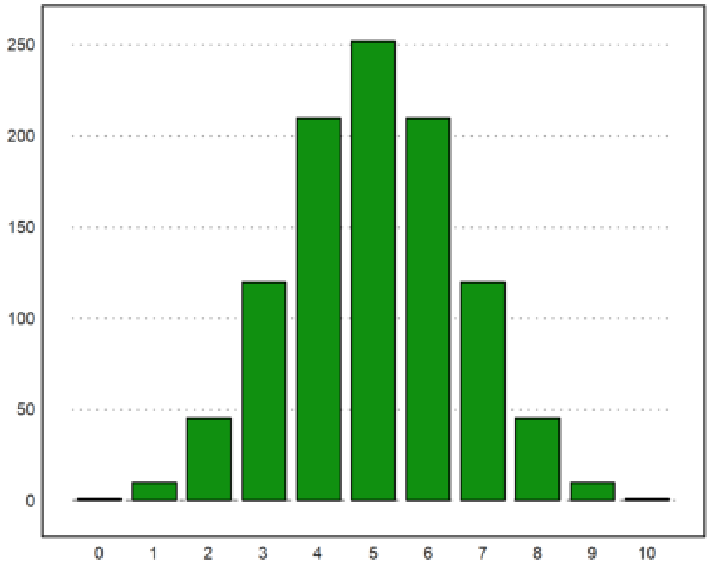
```
>plot2d(normal(10000),distribution=50):
```



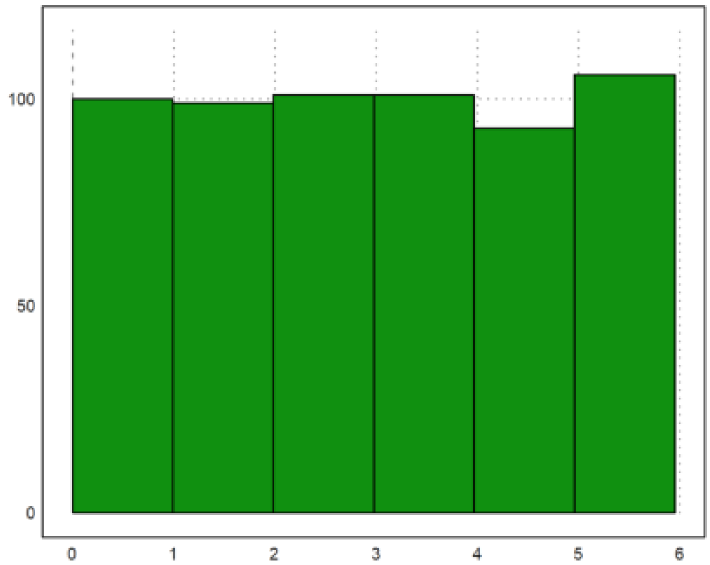

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




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

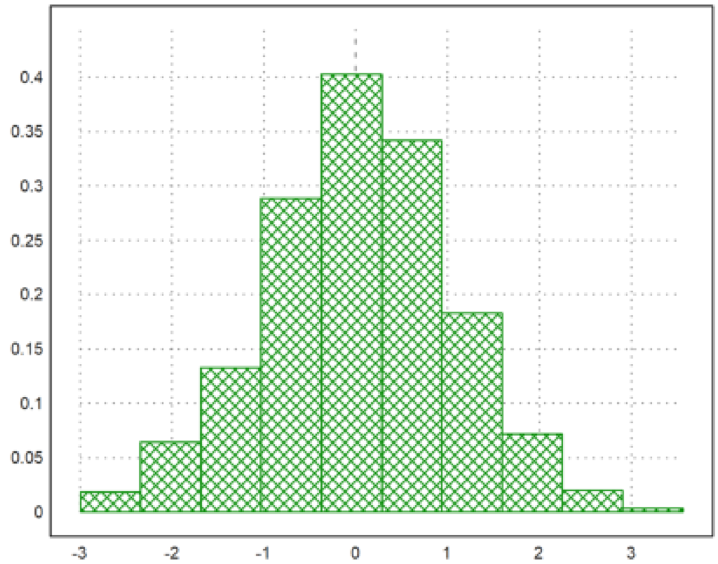



```
>plot2d(random(600)*6,histogram=6):
```



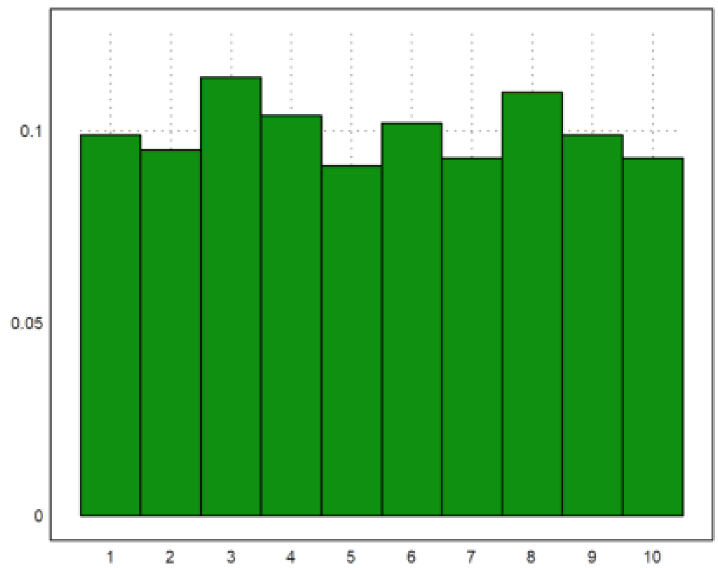
Untuk distribusi, ada parameter distribusi=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



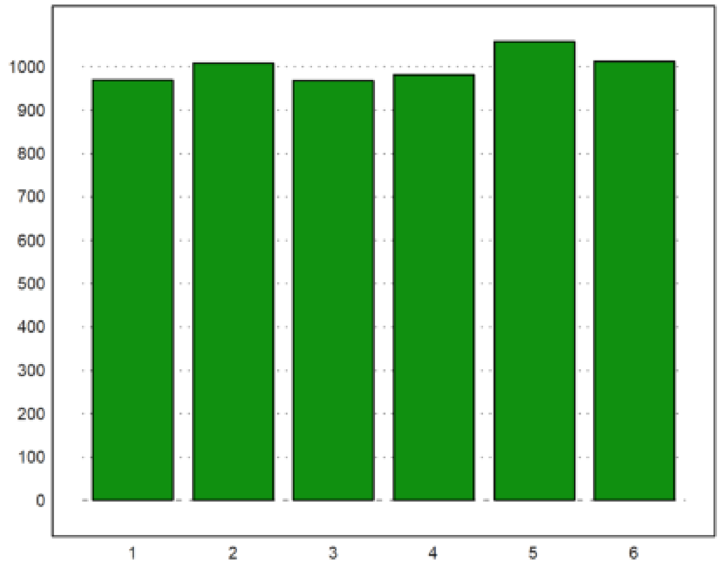
Dengan parameter even=true, ini akan menggunakan interval integer.

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

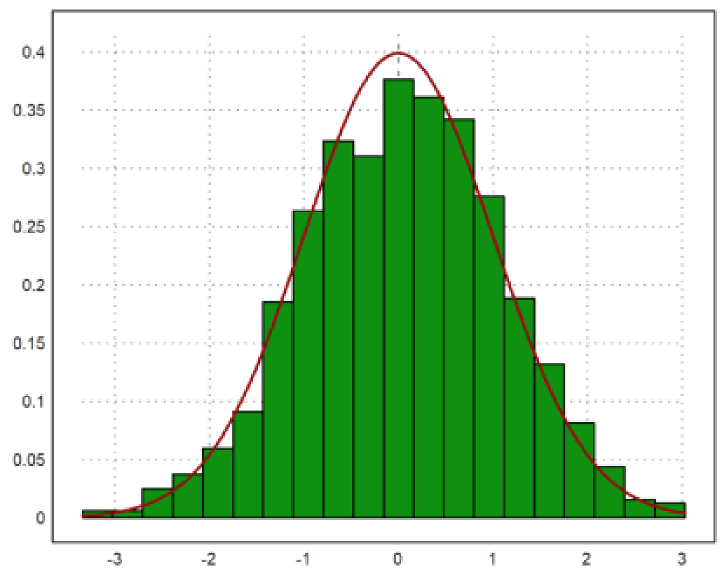


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

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

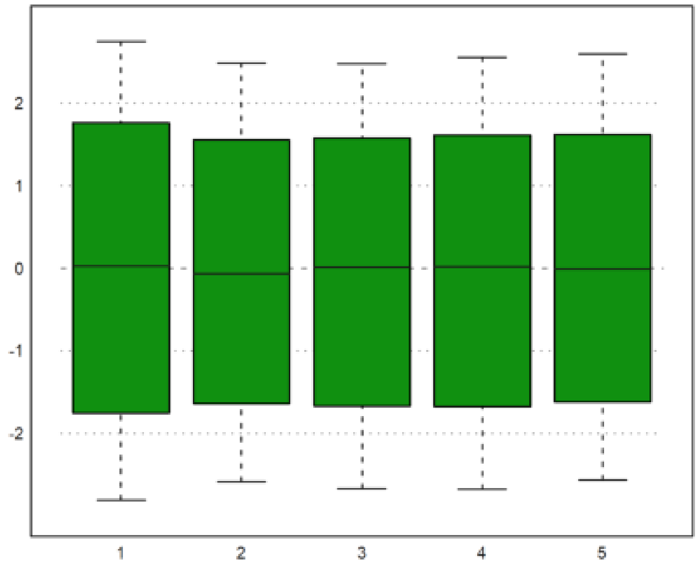



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

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



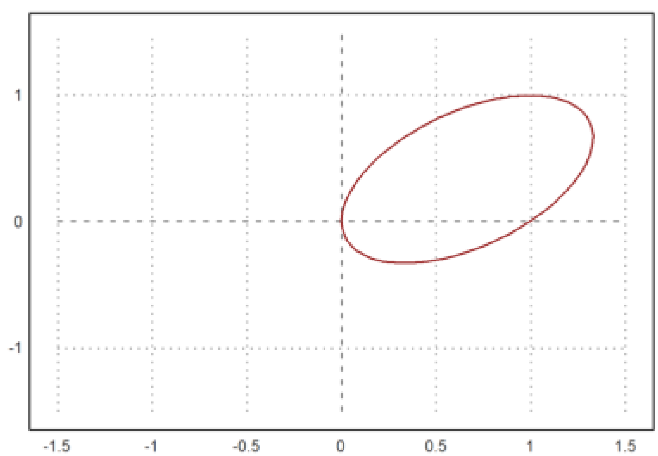
Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan f(x,y)=level, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada garis level nc, yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >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

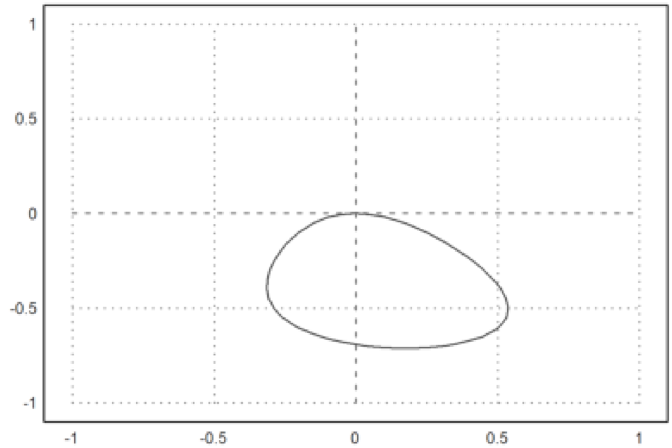
$$f(x, y) = c$$

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.

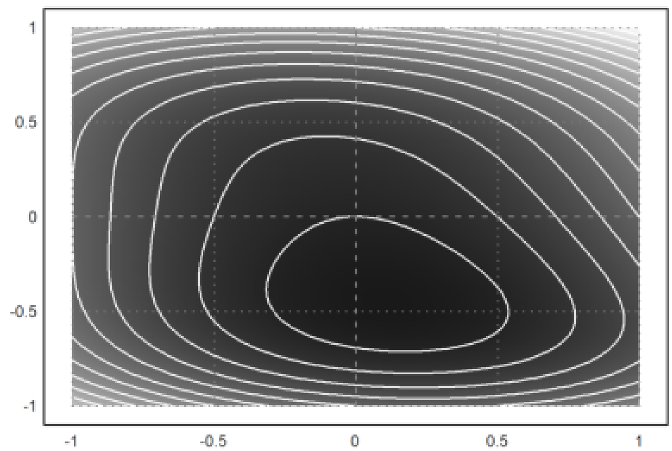
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



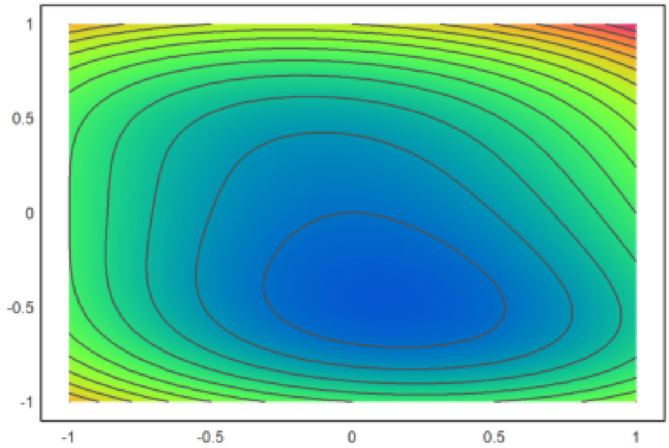
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```




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

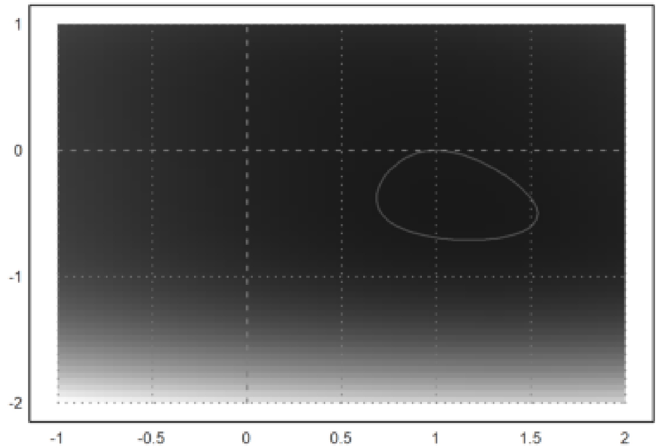



```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

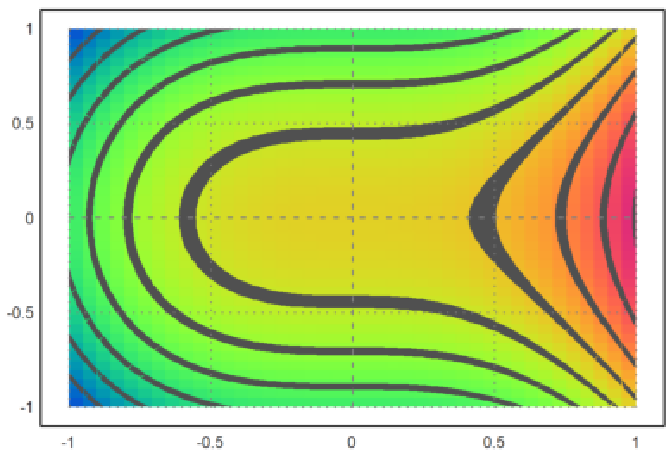


Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

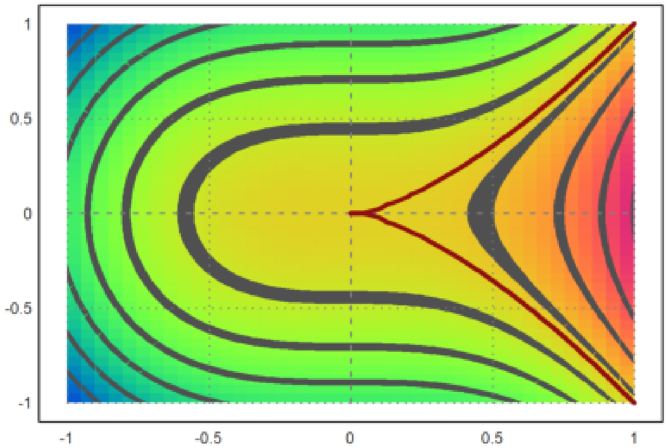
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



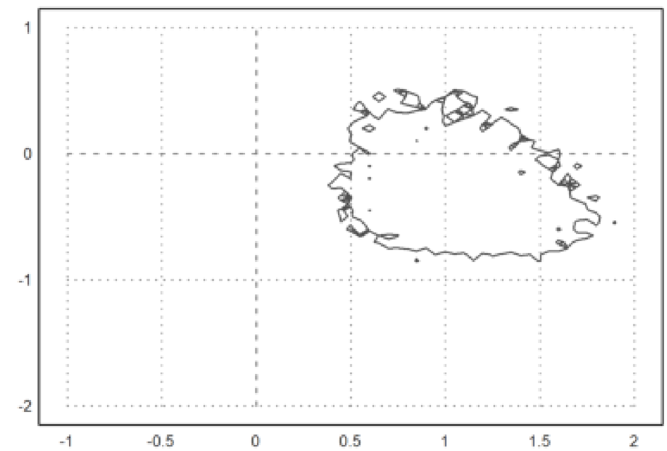

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



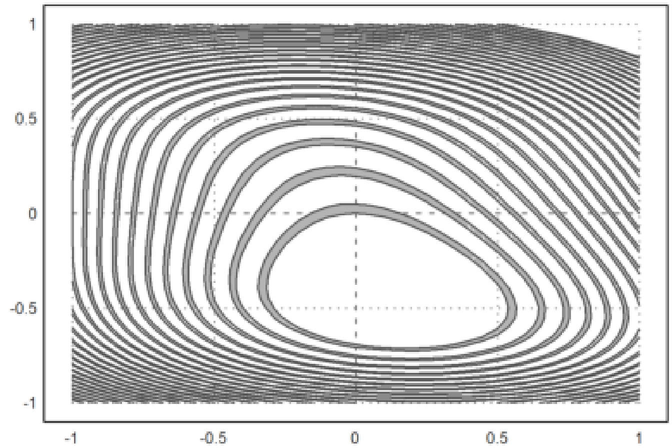
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



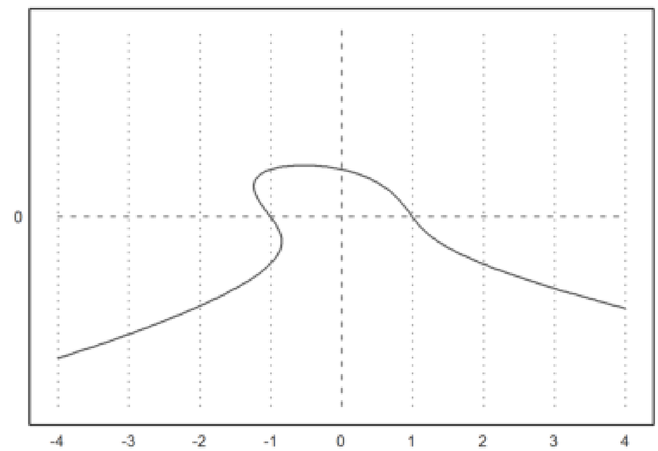

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



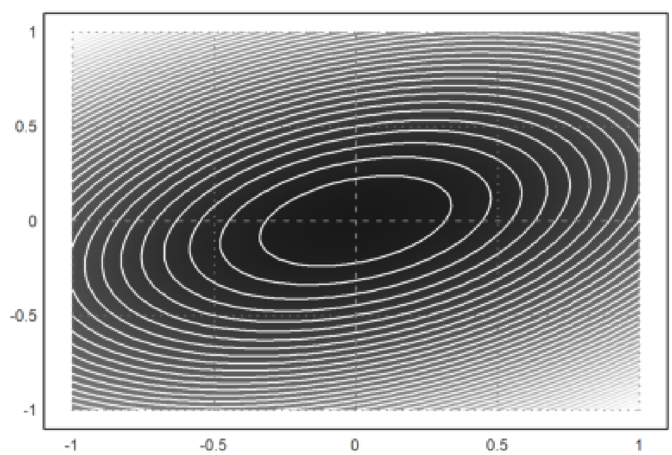

```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




```
>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):
```

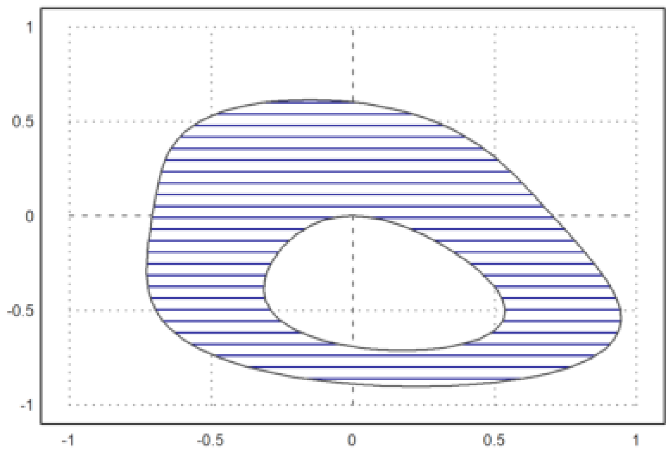


Juga dimungkinkan untuk mengisi set

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

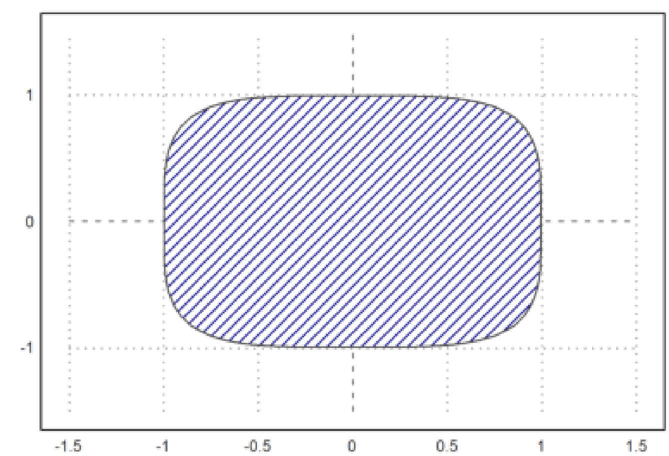
dengan rentang tingkat.
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

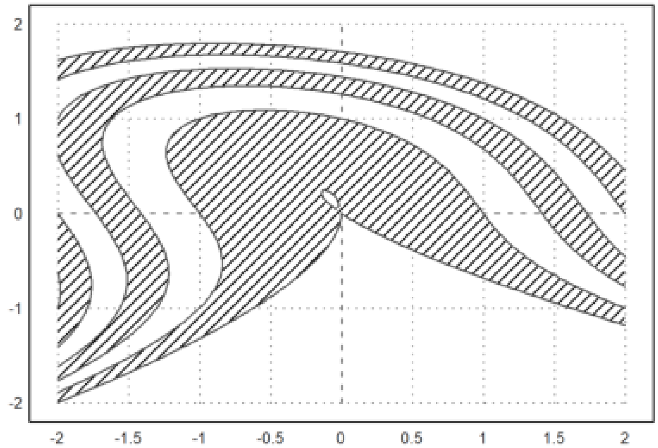


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

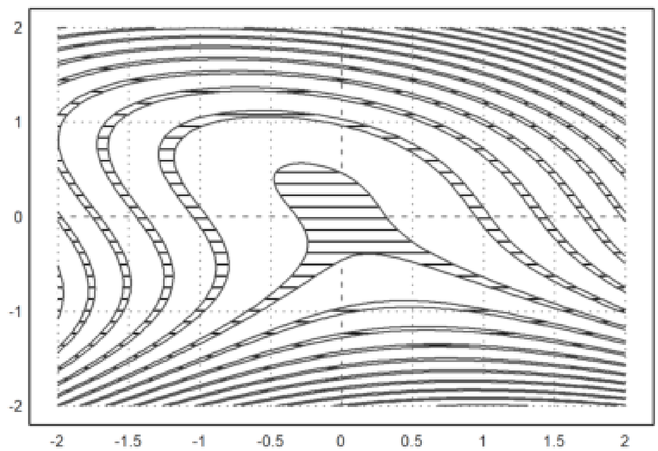
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



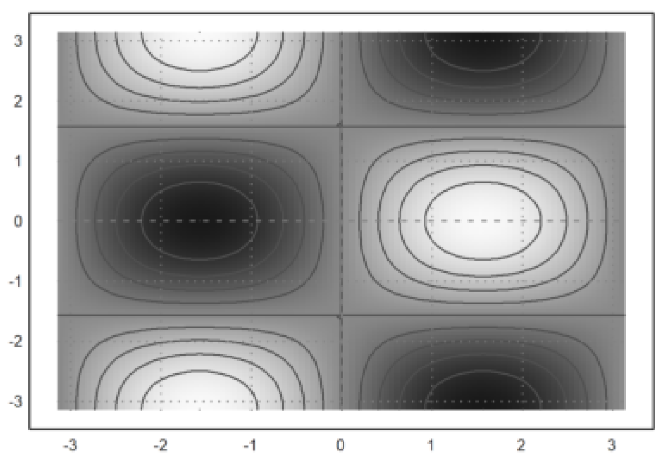
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




```
>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):
```

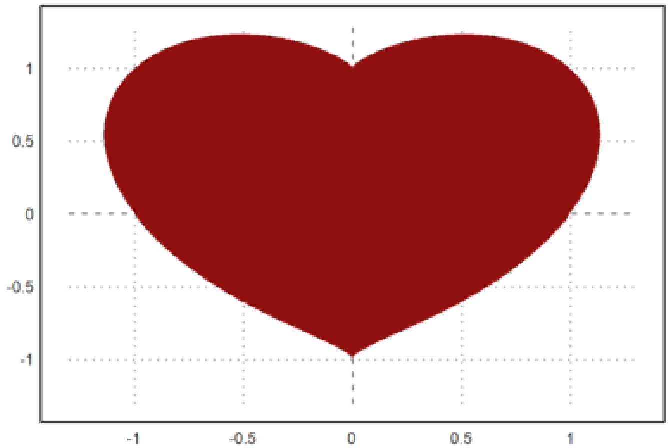


Dimungkinkan juga untuk menandai suatu wilayah

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

Ini dilakukan dengan menambahkan level dengan dua baris.

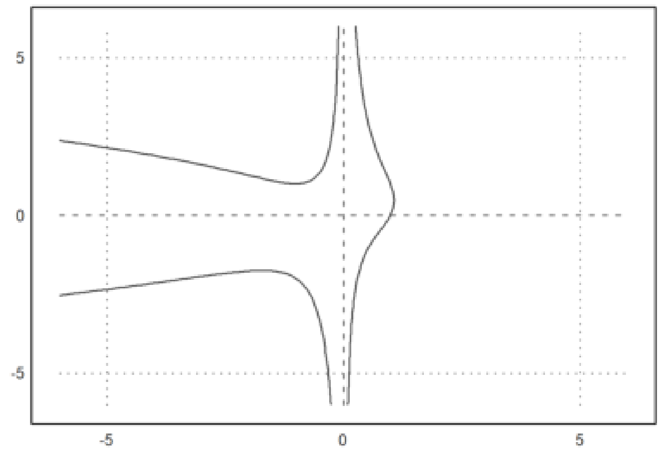
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```



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):
```

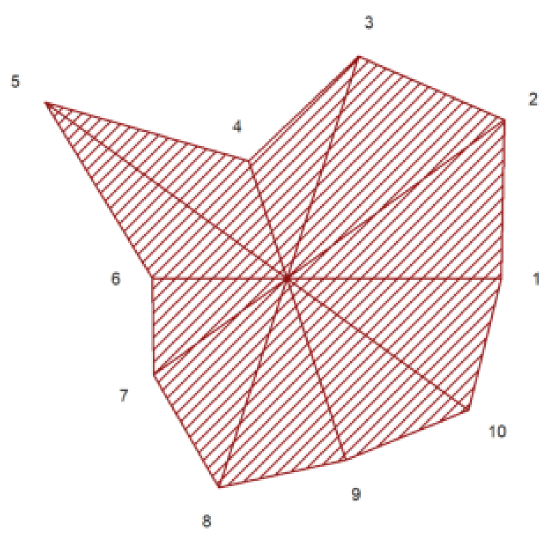



```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
    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):
```



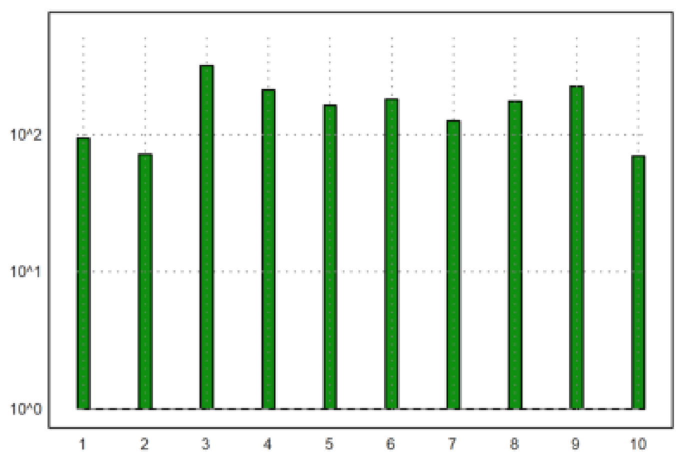
Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir.
Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```
>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.

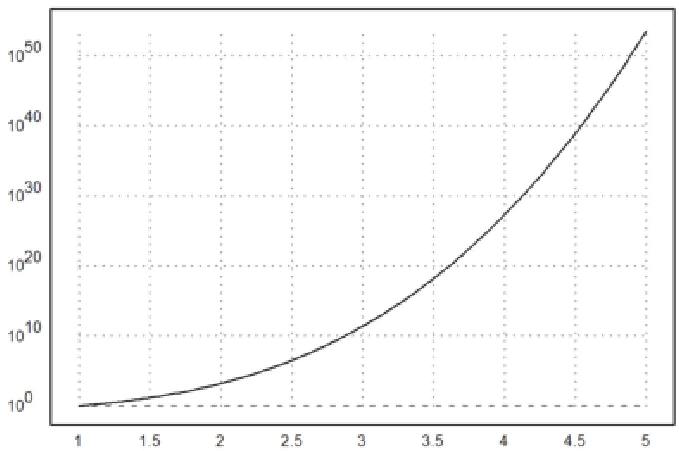
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

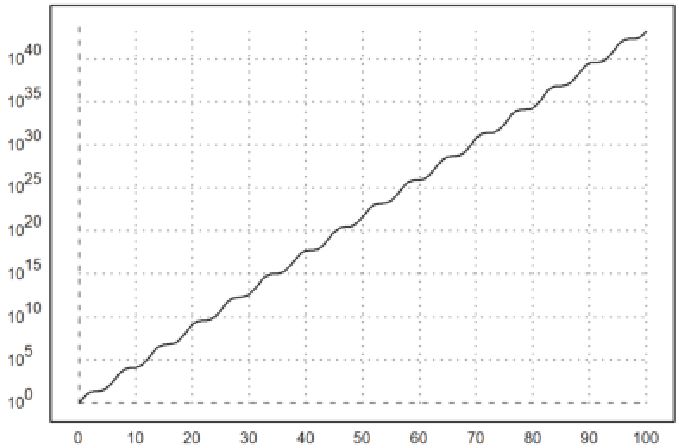
EMT menggunakan parameter "logplot" untuk skala logaritmik.
Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

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

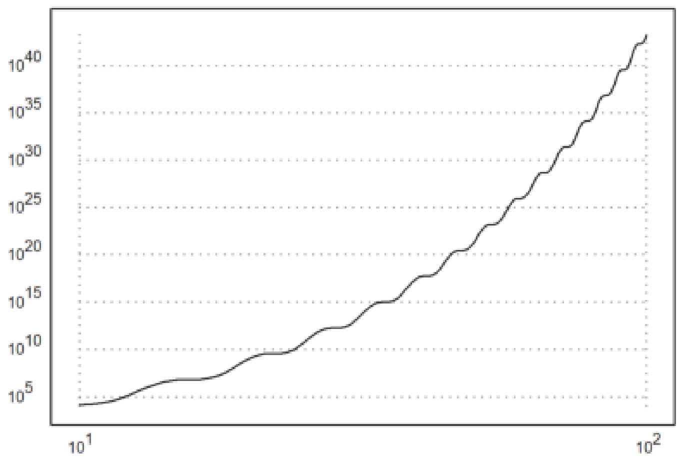
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



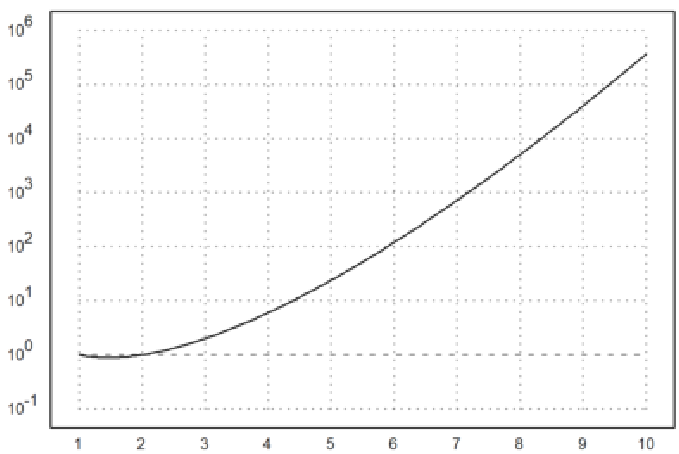

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



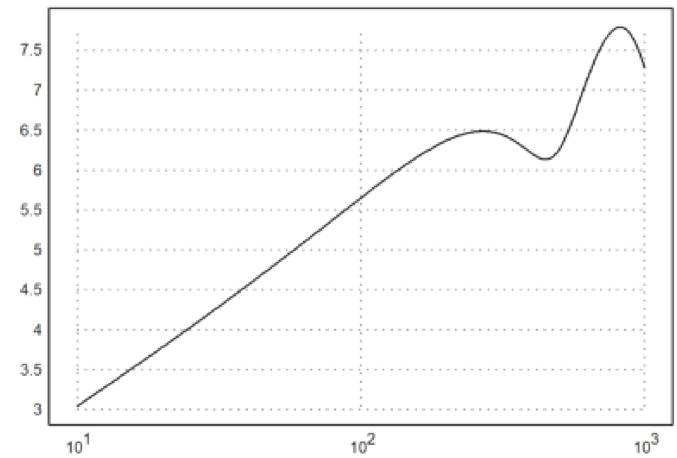

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




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

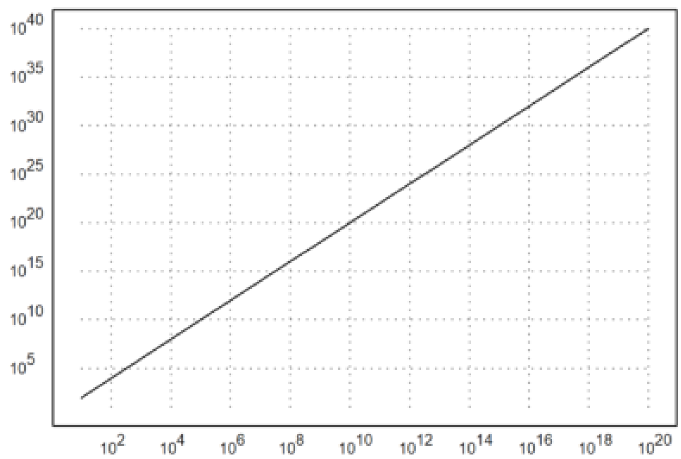


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



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
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 3

KB PEKAN 5: MENGGUNAKAN EMT UNTUK MENGAMBAR GRAFIK 3 DIMENSI (3D)

[a4paper,10pt]article eumat

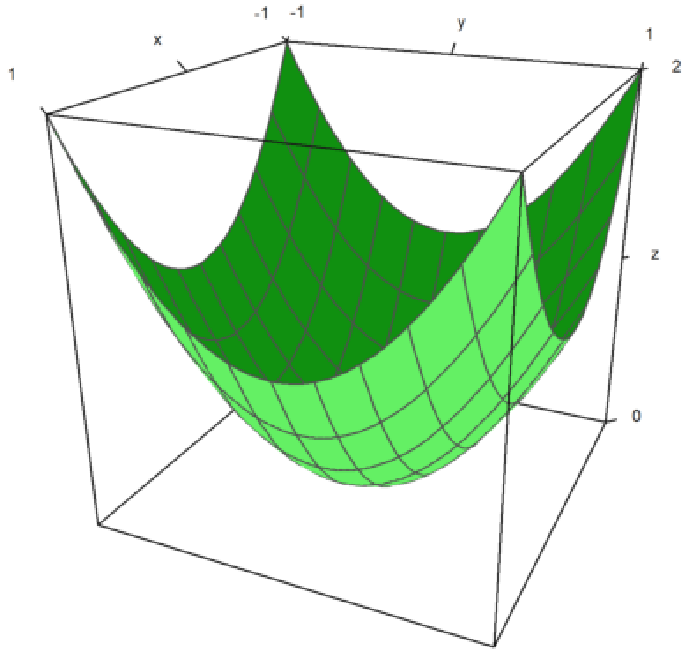
Nama: Akhmad Rizki khamid
Kelas: Matematika E
NIM: 22305141008

Menggambar Plot 3D dengan EMT

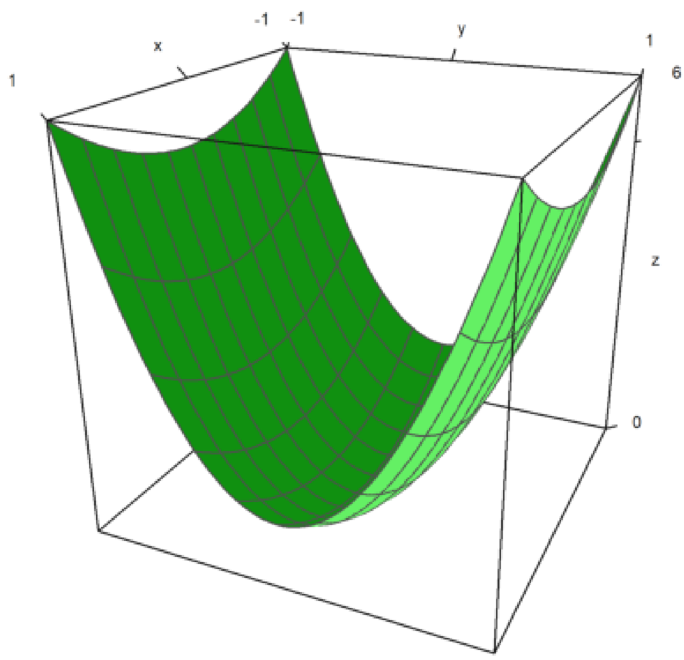
1. Menggambar Grafik Fungsi Dua Variabel * dalam Bentuk Ekspresi

Langsung
Fungsi Dua Variabel didefinisikan sebagai sebuah fungsi bernilai real dari dua variabel real, yakni fungsi f yang memadankan setiap pasangan
terurut (x,y) pada suatu himpunan D dari bidang dengan bilangan real
tunggal f (x,y).
Di dalam program numerik EMT, ekspresi adalah string. Jika ditandai sebagai simbolis, mereka akan mencetak melalui Maxima, jika tidak melalui EMT. Ekspresi dalam string digunakan untuk membuat plot dan banyak fungsi numerik. Untuk ini, variabel dalam ekspresi harus "x" dan "y".
Untuk grafik suatu fungsi, gunakan
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data.
contoh:

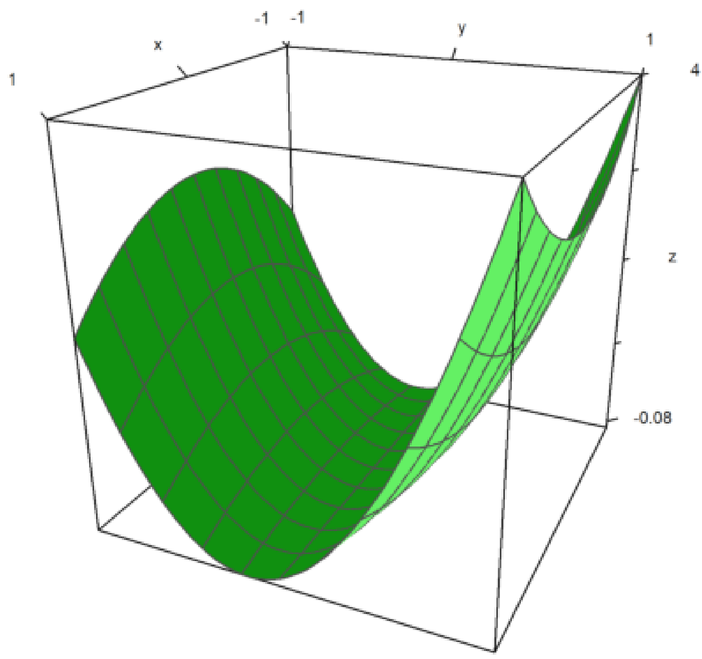
```
>plot3d("x^2+y^2"):
```



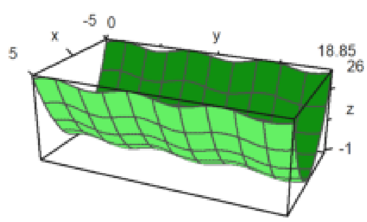

```
>plot3d("x^2+5*y^2"):
```



```
>plot3d("x^2*y+3*y^2"):
```




```
>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,6*pi):
```



1. aspect(1.5) mengatur aspek rasio pada grafik 3D.
2. plot3d("x^2+sin(y)",-5,5,0,6*pi) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
3. -5,5 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
4. 0,6*pi mengatur rentang sumbu y yang akan ditampilkan pada grafik.

Fungsi umum untuk plot 3D.

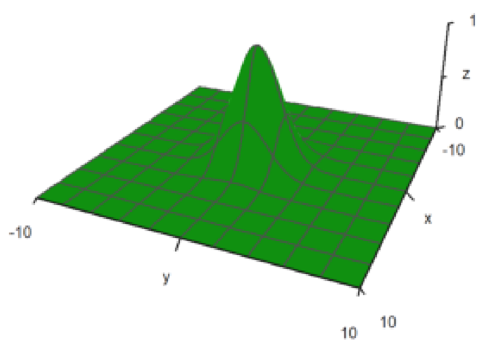
Fungsi plot3d (x, y, z, xmin, xmax, ymin, ymax, n, a, ..
b, c, d, r, scale, fscale, frame, angle, height, zoom, distance, ..)
Rentang plot untuk fungsi dapat ditentukan dengan
- a,b: rentang x
- c,d: rentang y
- r : persegi simetris di sekitar (0,0).
- n : jumlah subinterval untuk plot.
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.
- fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale).
- scale: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.
- frame: jenis bingkai (default 1).
Tampilan dapat diubah dengan berbagai cara.
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian pandangan dalam radian.
Nilai default dapat 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.
contoh:

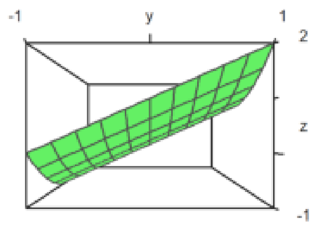
```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



1. exp(-(x^2+y^2)/5) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. r=10 mengatur jarak maksimum dari pusat grafik ke tepi grafik.
3. n=80 mengatur jumlah titik yang digunakan untuk membuat grafik.
4. fscale=4 mengatur faktor skala untuk warna.
5. scale=1.2 mengatur faktor skala untuk ukuran grafik.
6. frame=3 mengatur jenis bingkai yang digunakan untuk grafik.

Pada contoh berikut, sudut=0 dan tinggi=0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

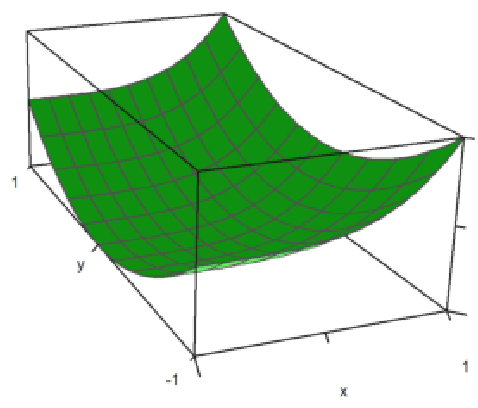
```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



1. x^2+y adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. distance=3 mengatur jarak pandang dari grafik.
3. zoom=1 mengatur faktor perbesaran grafik.
4. angle=pi/2 mengatur sudut pandang grafik dalam radian.
5. height=0 mengatur ketinggian pandangan dari grafik.

Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah 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):
```



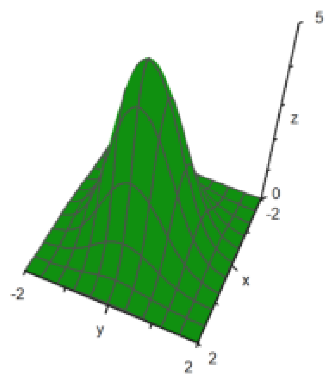
1. x^4+y^2 adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. a=0,b=1,c=-1,d=1 mengatur rentang sumbu x dan y yang akan ditampilkan pada grafik.
3. angle=-20° mengatur sudut pandang grafik dalam derajat.
4. height=20° mengatur ketinggian pandangan dari grafik dalam derajat.

5. center=[0.4,0,0] mengatur pusat pandangan dari grafik.
6. zoom=5 mengatur faktor perbesaran grafik.

Plotnya diskalakan agar sesuai dengan unit kubus untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung ukuran plot. Namun labelnya mengacu pada ukuran sebenarnya.

Jika Anda mematikannya dengan scale=false, Anda harus berhati-hati agar plot tetap masuk ke dalam jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengah.

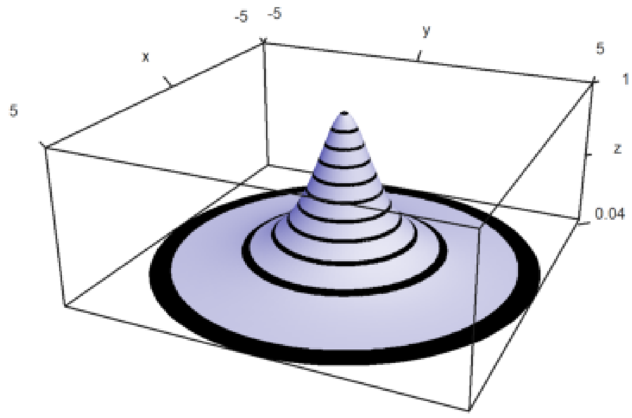
```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
>  center=[0,0,-2],frame=3):
```



1. 5*exp(-x^2-y^2) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. r=2 mengatur jarak maksimum dari pusat grafik ke tepi grafik.
3. <fscale mengatur faktor skala untuk warna.
4. <scale mengatur faktor skala untuk ukuran grafik.
5. distance=13 mengatur jarak pandang dari grafik.
6. height=50° mengatur ketinggian pandangan dari grafik dalam derajat.
7. center=[0,0,-2] mengatur pusat pandangan dari grafik.
8. frame=3 mengatur jenis bingkai yang digunakan untuk grafik.

Plot kutub juga tersedia. Parameter polar=true menggambar plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skalanya sendiri. Kalau tidak, fungsinya akan diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```

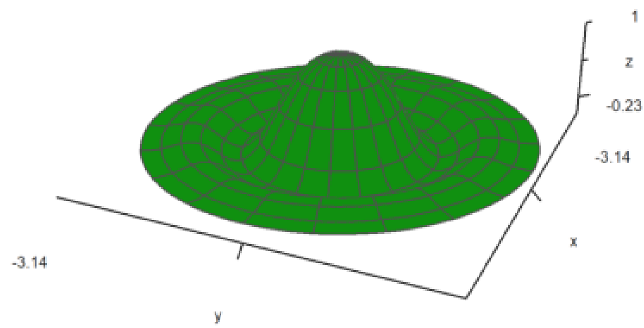


1. 1/(x^2+y^2+1) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. r=5 mengatur jarak maksimum dari pusat grafik ke tepi grafik.
3. polar mengatur tampilan grafik dalam koordinat polar.
4. fscale=2 mengatur faktor skala untuk warna.
5. hue mengatur skala warna yang digunakan pada grafik.
6. n=100 mengatur jumlah titik yang digunakan untuk membuat grafik.
7. zoom=4 mengatur faktor perbesaran grafik.
8. contour mengatur tampilan garis kontur pada grafik.
9. color=blue mengatur warna garis kontur pada grafik.

```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d"f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```

```
Function plot3d needs at least one argument!
Use: plot3d (x {, y: none, z: none, xmin: none, xmax: none, ...})
Error in:
plot3d"f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4) ...
      ^
```

```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```

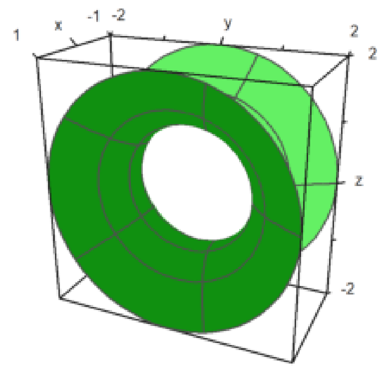


1. function f(r) := exp(-r/2)*cos(r) adalah fungsi matematika yang didefinisikan sebagai f(r) = e^(-r/2) * cos(r).
2. plot3d("f(x^2+y^2)",polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4) adalah perintah untuk membuat grafik 3D dari fungsi f(x^2+y^2).
3. polar mengatur tampilan grafik dalam koordinat polar.
4. scale=[1,1,0.4] mengatur faktor skala untuk ukuran grafik.
5. r=pi mengatur jarak maksimum dari pusat grafik ke tepi grafik.
6. frame=3 mengatur jenis bingkai yang digunakan untuk grafik.
7. zoom=4 mengatur faktor perbesaran grafik.

Parameter memutar memutar fungsi di x di sekitar sumbu x.

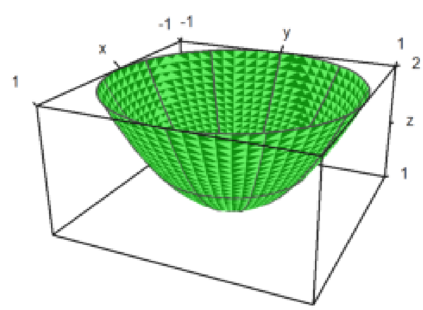
- rotate=1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



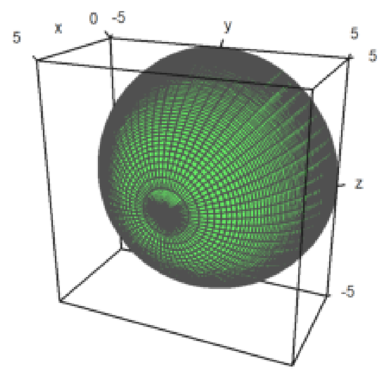
1. x^2+1 adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. a=-1,b=1 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3. rotate=true mengatur grafik agar dapat diputar secara interaktif.
4. grid=5 mengatur jumlah garis koordinat yang ditampilkan pada grafik.

```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



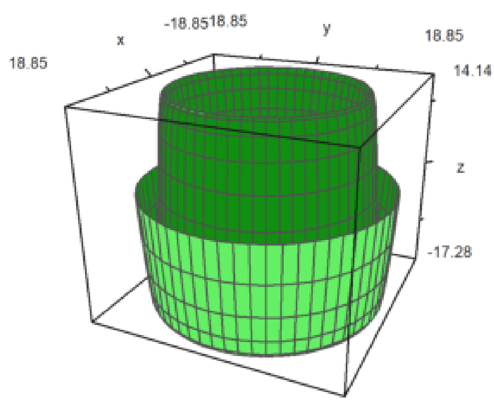
1. x^2+1 adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. a=-1,b=1 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3. rotate=2 mengatur grafik agar dapat diputar secara interaktif dengan menggunakan mouse.
4. grid=5 mengatur jumlah garis koordinat yang ditampilkan pada grafik.

```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```



1. sqrt(25-x^2) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. a=0,b=5 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3. rotate=1 mengatur grafik agar dapat diputar secara interaktif.

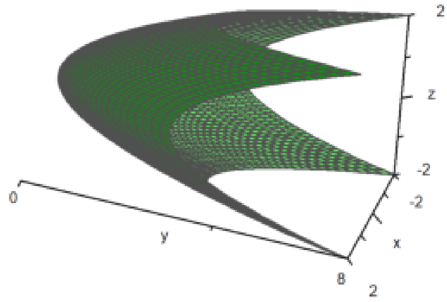
```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



1. x*sin(x) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. a=0,b=6pi mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3. rotate=2 mengatur grafik agar dapat diputar secara interaktif.

Berikut adalah plot dengan tiga fungsi.

```
>plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3):
```



1. x adalah fungsi matematika yang digunakan untuk menentukan nilai sumbu x pada grafik.
2. x^2+y^2 adalah fungsi matematika yang digunakan untuk menentukan nilai sumbu z pada grafik.
3. y adalah fungsi matematika yang digunakan untuk menentukan nilai sumbu y pada grafik.
4. r=2 mengatur jarak maksimum dari pusat grafik ke tepi grafik.
5. zoom=3.5 mengatur faktor perbesaran grafik.
6. frame=3 mengatur jenis bingkai yang digunakan untuk grafik.

2. Menggambar Grafik Fungsi Dua Variabel yang

* Rumusnya Disimpan dalam Variabel Ekspresi
Fungsi ini dapat memplot plot 3D dengan grafik fungsi dua variabel, permukaan berparameter, kurva ruang, awan titik, penyelesaian persamaan tiga variabel. Semua plot 3D bisa ditampilkan sebagai anaglyph.
fungsi plot3d (x, y, z, xmin, xmax, ymin, ymax, n, a
Parameter
x : ekspresi dalam x dan y
x,y,z : matriks koordinat suatu permukaan
x,y,z : ekspresi dalam x dan y untuk permukaan parametrik
x,y,z : ekspresi dalam x untuk memplot kurva ruang
xmin,xmax,ymin,ymax :

```
x,y batas ekspresi
```

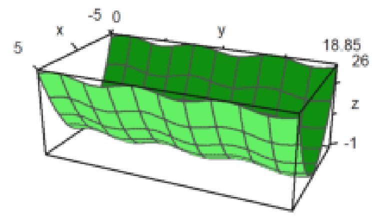
contoh:
ekspresi dalam string

```
>expr := "x^2+sin(y)"
```

```
x^2+sin(y)
```

plot ekspresi

```
>plot3d(expr,-5,5,0,6*pi):
```



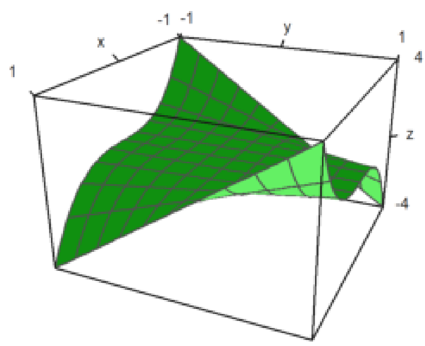
1. x^2+sin(y) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2. -5,5 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3. 0,6*pi mengatur rentang sumbu y yang akan ditampilkan pada grafik.

contoh 1:

```
>
>expr := "4*x^3*y"
```

```
4*x^3*y
```

```
>aspect(1.5); plot3d(expr):
```



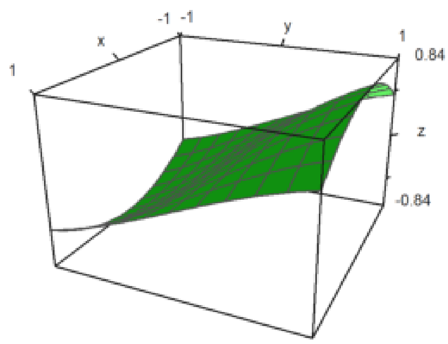
1. aspect(2) mengatur aspek rasio pada grafik 3D.
2. plot3d(expr) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.

contoh 2:

```
>expr := "cos(x)*sin(y)"
```

```
cos(x)*sin(y)
```

```
>plot3d(expr):
```

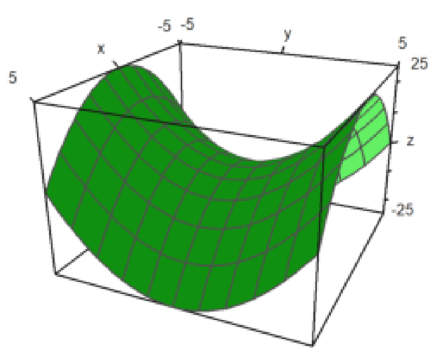


contoh 3:

```
>expr := "y^2-x^2"
```

```
y^2-x^2
```

```
>aspect(1.5); plot3d(expr,-5,5,-5,5):
```



3. Menggambar Grafik Fungsi Dua Variabel yang

* Fungsinya Didefinisikan sebagai Fungsi Numerik

Fungsi Dua Variabel

Fungsi dua variabel adalah sebuah fungsi yang bernilai real dari dua variabel real. Fungsi ini memadankan setiap pasangan terurut (x,y) pada suatu himpunan D dari bidang dengan bilangan real tunggal f(x,y). Dalam matematika, fungsi dua variabel atau lebih digunakan untuk menggambarkan hubungan antara dua atau lebih variabel.

Fungsi Numerik

Fungsi numerik adalah suatu fungsi matematika yang menghasilkan nilai numerik sebagai output-nya. Fungsi ini dapat dinyatakan dalam bentuk persamaan matematika atau algoritma komputasi.
Contoh:
Fungsi

$$f(x, y) = 5x + y$$

Misal input nilai x=2 dan y=3, maka akan dihasilkan nilai z yaitu

$$z = f(x, y) = 5(2) + 3 = 10 + 3 = 13$$

Gambar Grafik Fungsi

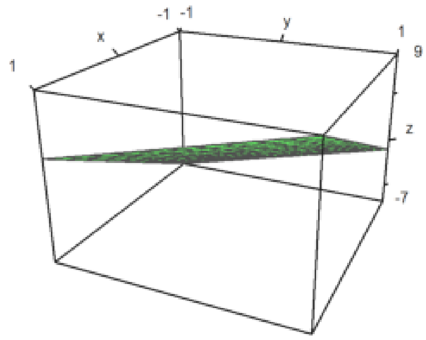
Fungsi satu baris numerik didefinisikan oleh ":=".
Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsi nya didefinisikan sebagai fungsi numerik dalam plot3d:
1. Buat fungsi numerik yang akan digunakan untuk memvisualisasikan data.
function f(x,y):=ax+by
dimana a dan b adalah konstanta
2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi numerik.
plot3d("f"):

Contoh

Fungsi matematika f(x,y) dapat digambarkan dalam bentuk grafik tiga dimensi menggunakan perintah plot3d. Berikut adalah contoh penggunaan perintah plot3d untuk menggambarkan fungsi tersebut:
1. Fungsi Linear Dua Variabel

$$f(x, y) = 5x + 3y + 1$$

```
>function f(x,y):= 5*x+3*y+1
>plot3d("f"):
```



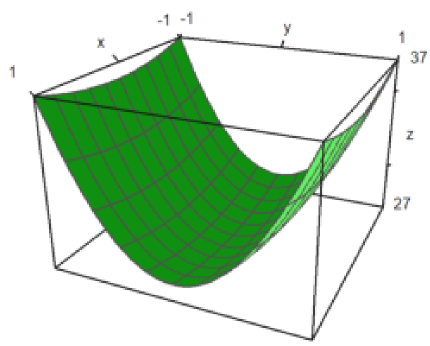
- Fungsi f(x,y) didefinisikan sebagai 5x+3y+1.
- Perintah "plot3d("f")" digunakan untuk memplot grafik 3D dari fungsi f(x,y) menggunakan fungsi plot3d di EMT.
- Grafik yang dihasilkan akan menampilkan fungsi dalam tiga dimensi, dengan sumbu x dan y mewakili variabel masukan dan sumbu z mewakili nilai keluaran fungsi. Grafik akan menunjukkan bentuk fungsi dan perubahannya seiring dengan perubahan variabel masukan.

---

2. Fungsi Kuadrat Dua Variabel

$$f(x, y) = x^2 + 3y^2 + 21$$

```
>function f(x,y):= x^2+(3*y)^2+27
>plot3d("f"):
```



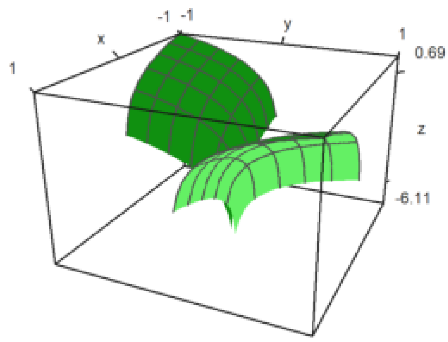
- Perintah "function f(x,y):= x^2+(3*y)^2+27" berarti mendefinisikan fungsi matematika f(x,y) sebagai x pangkat 2 ditambah 3 kali y pangkat 2 ditambah 27.
- Perintah "plot3d("f")" berarti membuat grafik tiga dimensi dari fungsi f(x,y) yang telah didefinisikan sebelumnya.

---

3. Fungsi Logaritma Dua Variabel

$$f(x, y) = \log(2xy)$$

```
>function f(x,y):= log((2*x)*y)
>plot3d("f"):
```



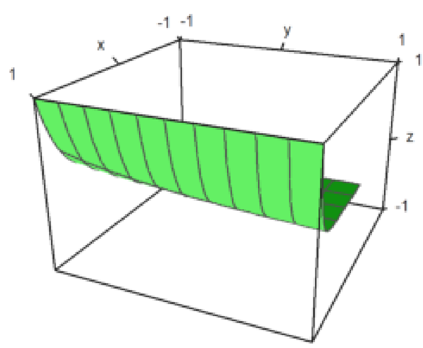
- Input yang diberikan adalah fungsi matematika dua variabel, f(x,y), yang didefinisikan sebagai logaritma hasil kali 2x dan y.
- Perintah "plot3d("f")" digunakan untuk memplot grafik fungsi f(x,y) dalam ruang tiga dimensi.

---

4. Fungsi Eksponen Dua Variabel

$$f(x, y) = x^{5y+10}$$

```
>function f(x,y):= x^(5*y+10)
>plot3d("f"):
```



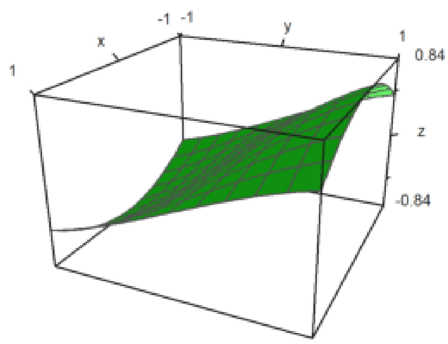
- Perintah ‘fungsi f(x,y):= x^(5y+10)‘ adalah fungsi matematika dua variabel ‘x‘ dan ‘y‘ dan dengan rumus x^(5y+10)
- Perintah ‘plot3d("f")‘ digunakan untuk memplot fungsi dalam ruang tiga dimensi. Plot yang dihasilkan akan menampilkan nilai fungsi sebagai permukaan pada bidang x-y, dengan tinggi permukaan mewakili nilai fungsi pada titik tersebut.

---

5. Fungsi Trigonometri Dua Variabel

$$f(x,y) = cos(x) + sin(y)$$

```
>function f(x,y):= cos(x)*sin(y)
>plot3d("f"):
```



- Perintah "function f(x,y):= cos(x)*sin(y)" adalah perintah untuk mendefinisikan fungsi matematika f(x,y) yang menghasilkan nilai cosinus dari x dikalikan dengan sinus dari y.
- Perintah "plot3d("f")" adalah perintah untuk membuat grafik tiga dimensi dari fungsi f(x,y) yang telah didefinisikan sebelumnya.

4. Menggambar Grafik Fungsi Dua Variabel yang

* Fungsinya Didefinisikan sebagai Fungsi Simbolik

Fungsi Simbolik

Fungsi simbolik adalah fungsi yang dinyatakan dengan menggunakan simbol-simbol matematika, seperti huruf dan operasi matematika, daripada menggunakan angka konkret atau ekspresi numerik. Fungsi simbolik sering digunakan untuk menggambarkan hubungan antara variabel-variabel matematika dalam bentuk yang lebih umum dan abstrak.
Contoh fungsi simbolik yang umum adalah:

$$g(x,y) = 2x + y$$

Dalam contoh di atas, g(x) adalah fungsi simbolik yang mengaitkan setiap nilai x dengan hasil dari ekspresi matematika 2x + 3. Fungsi ini dapat digunakan untuk menghitung nilai fungsi untuk berbagai nilai x.

Perbedaan Fungsi Numerik dan Fungsi Simbolik

1. Fungsi Numerik
Fungsi numerik dinyatakan dalam bentuk yang lebih konkret menggunakan angka-angka nyata.
Contoh fungsi numerik adalah

$$g(x, y) = 2x + y + 3$$

dimana kita memberikan nilai numerik kepada "x dan y"
misalnya, x = 5 dan y = 2, maka hasilnya adalah angka konkret yaitu g(5,2) = 15
2. Fungsi Simbolik
Fungsi simbolik dinyatakan menggunakan simbol-simbol matematika seperti huruf (variabel) dan operasi matematika.
Contoh fungsi simbolik adalah

$$g(x, y) = 2x + y + 3$$

"g" adalah simbol fungsi
"x,y" adalah variabel,
2x + 3 adalah ekspresi matematika yang menggambarkan hubungan antara "x,y" dan hasil fungsi.

Gambar Grafik Fungsi

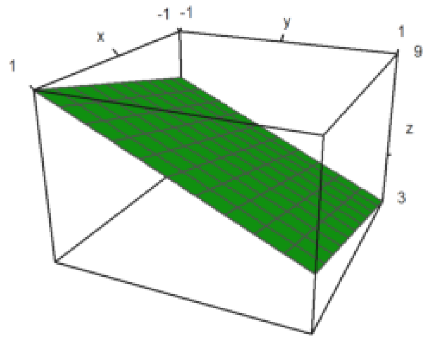
Fungsi satu baris simbolik didefinisikan oleh "&=".
Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsi nya didefinisikan sebagai fungsi simbolikdalam plot3d:
1. Buat fungsi simbolik yang akan digunakan untuk memvisualisasikan data.
function g(x,y):= ax+by;
dimana a dan b adalah konstanta
2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi numerik.
plot3d("g"):
3. Menentukan rentang variabel
misal
plot3d("g(x,y)",-10,10,-5,5):
dengan batasan x dari -10 hingga 10 dan batasan y dari -5 hingga 5

Contoh

1. Fungsi Linear Dua Variabel

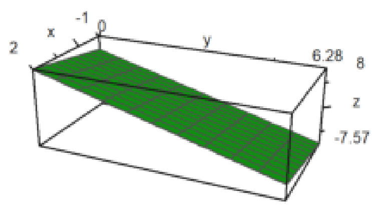
$$g(x, y) = x - 2y + 6$$

```
>function g(x,y)&= x-2*y+6;
>plot3d("g(x,y)"):
```



- Fungsi g(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus x - 2y + 6.
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,2,0,2*pi):
```



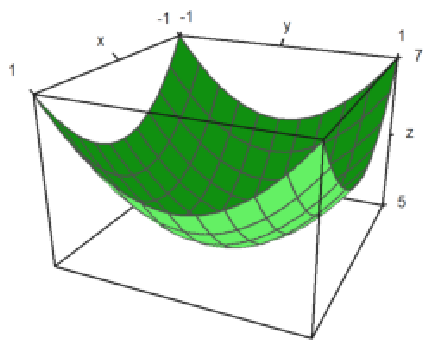
- Perintah "plot3d("g(x,y)",-1,2,0,2*pi)" adalah perintah untuk menggambar grafik fungsi tiga dimensi "g(x,y)" pada rentang x dari -1 hingga 2 dan rentang y dari 0 hingga 2pi.

---

2. Fungsi Kuadrat Dua Variabel

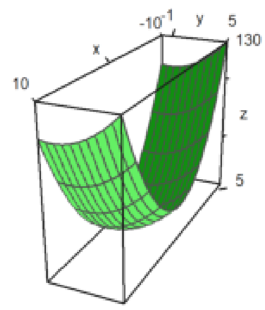
$$g(x, y) = x^2 + y^2 + 5$$

```
>function g(x,y)&= x^2+y^2+5;
>plot3d("g(x,y)"):
```



- Fungsi g(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus x^2+y^2+5
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-10,10,-1,5):
```



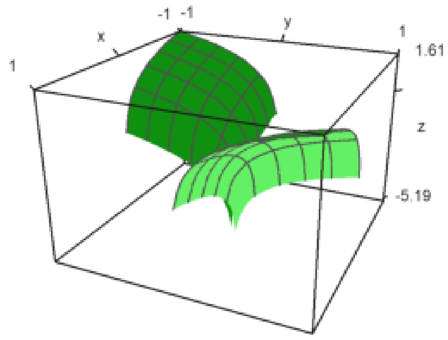
- Perintah "plot3d("g(x,y)",-10,10,-1,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi g(x,y) pada rentang x dari -10 hingga 10 dan rentang y dari -1 hingga 5

---

3. Fungsi Logaritma Dua Variabel

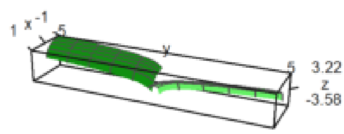
$$g(x, y) = \log(xy5)$$

```
>function g(x,y)&= log(x*y*5);
>plot3d("g(x,y)"):
```



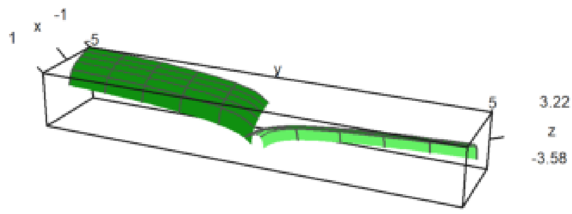
- Fungsi g(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus logaritma x dikalikan y dikalikan 5
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,1,-5,5):
```



- Perintah "plot3d("g(x,y)",-1,1,-5,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi g(x,y) pada rentang x dari -1 hingga 1 dan rentang y dari -5 hingga 5

```
>plot3d("g(x,y)",-1,1,-5,5,zoom=4.5):
```



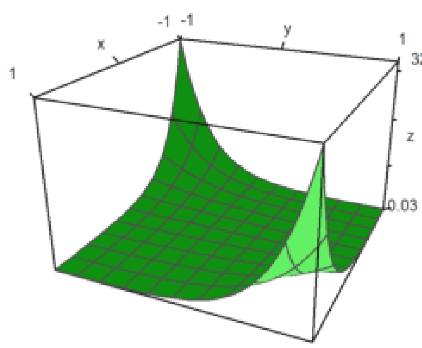
- plot3d: perintah untuk membuat grafik 3D.
- "g(x,y)": fungsi matematika yang akan digunakan untuk membuat grafik.
- -1,1: rentang nilai variabel x yang akan digunakan dalam grafik.
- -5,5: rentang nilai variabel y yang akan digunakan dalam grafik.
- zoom=4.5: perintah untuk memperbesar tampilan grafik dengan faktor 4.5.

---

4. Fungsi Eksponen Dua Variabel

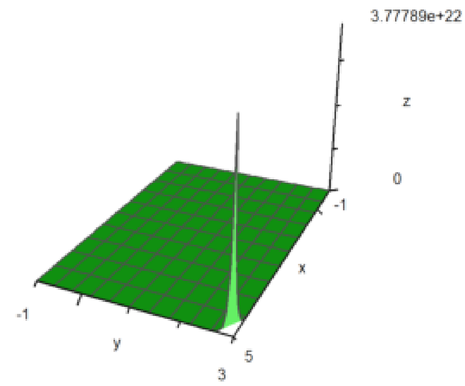
$$g(x, y) = 2^{xy5}$$

```
>function g(x,y)&= 2^(x*y*5);
>plot3d("g(x,y)"):
```



- Fungsi g(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus 2^(xy5)
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3):
```



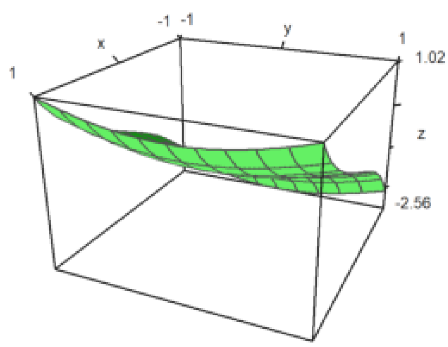
- Peintah plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3) adalah perintah untuk membuat plot tiga dimensi dari fungsi ‘g(x,y)‘ dengan batas ‘x‘ dari ‘-1‘ hingga ‘5‘ dan batas ‘y‘ dari ‘-1‘ hingga ‘3‘.
- plot3d: perintah untuk membuat plot tiga dimensi.
- "g(x,y)"‘: fungsi yang akan diplot.
- (-1,5): batas ‘x‘ dari ‘-1‘ hingga ‘5‘.
- (-1,3): batas ‘y‘ dari ‘-1‘ hingga ‘3‘.
- frame=3: menampilkan frame nomor 3.
- zoom=3: memperbesar tampilan plot sebanyak 3 kali.

---

5. Fungsi Trigonometri Dua Variabel

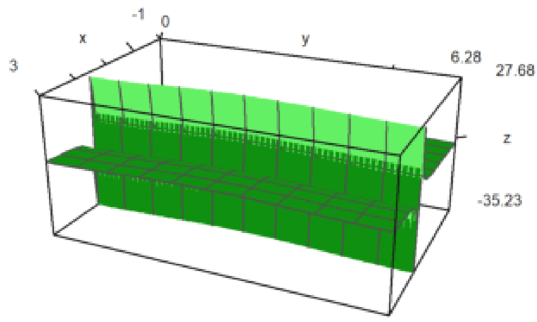
$$g(x, y) = tan(x) - cot(y)$$

```
>function g(x,y)&= tan(x)-cos(y);
>plot3d("g(x,y)"):
```



- Fungsi g(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus tan(x)-cos(y)
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,3,0,2*pi,frame=1,zoom=3.5):
```



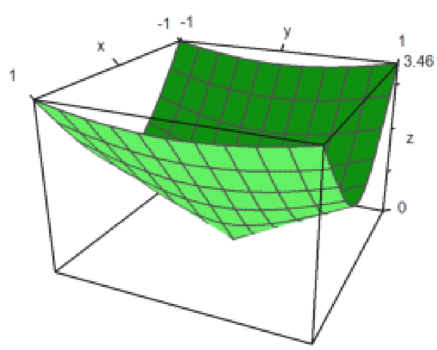
- Peintah plot3d("g(x,y)",-1,3,0,2*pi,frame=5,zoom=3) adalah perintah untuk membuat plot tiga dimensi dari fungsi ‘g(x,y)‘ dengan batas ‘x‘ dari ‘-1‘ hingga ‘3‘ dan batas ‘y‘ dari ‘0‘ hingga ‘2pi‘.
- plot3d: perintah untuk membuat plot tiga dimensi.
- "g(x,y)"‘: fungsi yang akan diplot.
- (-1,3): batas ‘x‘ dari ‘-1‘ hingga ‘3‘.
- (0,2pi): batas ‘y‘ dari ‘0‘ hingga ‘2pi‘.
- frame=1: menampilkan frame nomor 1.
- zoom=3.5: memperbesar tampilan plot sebanyak 3.5 kali.

---

6. Fungsi Akar Kuadrat

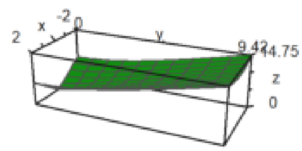
$$P(x,y) = \sqrt{10x^2 + 2y^2}$$

```
>function P(x,y) &= sqrt(10*x^2+2*y^2);
>plot3d("P(x,y)"):
```



- Fungsi P(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus akar kuadrat dari 10x^2+2y^2
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("P(x,y)",-2,2,0,3*pi,frame=5,zoom=2,scale=1):
```



- P(x,y): Merupakan fungsi yang akan digambarkan dalam grafik tiga dimensi.
- (-2,2): Merupakan rentang nilai dari sumbu x yang akan digunakan dalam grafik.
- (0,3pi): Merupakan rentang nilai dari sumbu y yang akan digunakan dalam grafik. Nilai pi dikalikan dengan 3 agar rentang nilai y mencakup tiga putaran lingkaran penuh.
- frame=5: Menentukan nomor bingkai (frame) yang akan digunakan dalam animasi grafik.
- zoom=2: Menentukan faktor pembesaran grafik. Dengan memperbesar tampilan, kita dapat melihat detail yang lebih kecil pada plot.
- scale=1: Menentukan skala grafik. Dengan mengatur skala, kita dapat mengubah jarak antara titik-titik pada sumbu tersebut.

1. aspect(1.5) mengatur aspek rasio pada grafik 3D.
2. plot3d(expr,-5,5,-5,5) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
3. -5,5 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
4. -5,5 mengatur rentang sumbu y yang akan ditampilkan pada grafik.

Menggambar Data x, y, z

* pada ruang Tiga Dimensi (3D)
Definisi

```
Menggambar data pada ruang tiga dimensi (3D) adalah proses
```

visualisasi data yang mengubah informasi dalam tiga dimensi, yaitu panjang, lebar, dan tinggi, menjadi representasi visual yang dapat dipahami dan dianalisis.
Tujuan:

```
Tujuan dari menggambar data 3D adalah untuk membantu pemahaman dan
```

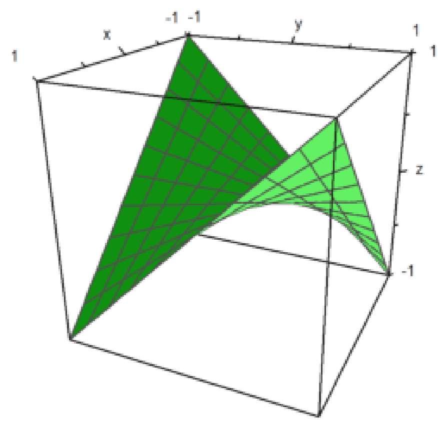
interpretasi data yang lebih baik, terutama ketika data tersebut memiliki komponen yang tidak dapat direpresentasikan dengan baik dalam dua dimensi.
Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x-, y- dan z, atau tiga fungsi atau ekspresi fx(x,y), fy(x,y), fz(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) melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.
Kita 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 dapat menandai daerah, dalam kasus kita daerah kutub.

Contoh 1

```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```

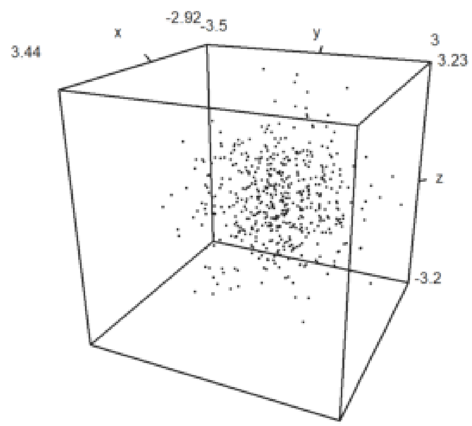


Baris pertama kode "t=-1:0.1:1" membuat vektor baris t yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Baris kedua "s=(-1:0.1:1)'" membuat vektor kolom s yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Operator transpose ' digunakan untuk mengubah vektor baris t menjadi vektor kolom.
Baris ketiga "plot3d(t,s,ts,grid=10)" membuat plot tiga dimensi dari fungsi f(x,y) = xy pada domain [-1,1] x [-1,1]. Plot dibuat menggunakan fungsi plot3d, yang mengambil tiga argumen: koordinat x, y, dan z dari titik-titik yang akan diplot. Dalam hal ini, koordinat x diberikan oleh vektor t, koordinat y diberikan oleh vektor s, dan koordinat z diberikan oleh hasil perkalian t dan s, yaitu ts. Parameter grid diatur menjadi 10, yang menunjukkan jumlah garis grid yang akan ditampilkan pada plot.

Contoh 2

Tentu saja, titik cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik-titik tersebut.
Gayanya sama seperti di plot2d dengan points=true;

```
>n=500;...
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```



Kode "n=500; plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".")" digunakan untuk membuat plot tiga dimensi dari tiga vektor normal yang dihasilkan secara acak dengan menggunakan fungsi "normal()" pada Euler Math Toolbox (EMT). Parameter "n=500" menunjukkan bahwa setiap vektor normal memiliki 500 elemen. Parameter "points=true" digunakan untuk menampilkan titik-titik pada plot, sedangkan parameter "style='.'" digunakan untuk mengatur gaya titik pada plot menjadi titik bulat.

Contoh 3

```
Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.
```

```
Dalam contoh berikut, kita membuat tampilan bayangan 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 mendistorsi 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):
```

Kode ini terdiri dari beberapa baris. Baris pertama "t=linspace(0,2pi,320)" membuat vektor t yang berisi 320 nilai yang sama terdistribusi secara merata antara 0 dan 2p. Baris kedua "s=linspace(-pi/2,pi/2,160)'" membuat vektor s yang berisi 160 nilai yang sama terdistribusi secara merata antara -p/2 dan p/2. Operator transpose ' digunakan untuk mengubah vektor baris s menjadi vektor kolom.

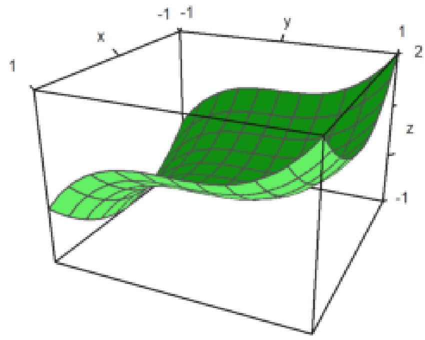
Baris ketiga "d=1+0.2*(cos(4t)+cos(8s))" membuat vektor d yang berisi nilai dari 1 + 0.2 * (cos(4t) + cos(8s)). Baris keempat "plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1,light=[1,0,1],frame=0,zoom=5)" membuat plot tiga dimensi dari fungsi f(x,y) = 2x^2 + y^3. Plot dibuat menggunakan fungsi plot3d, yang mengambil empat argumen: koordinat x, y, dan z dari titik-titik yang akan diplot, serta beberapa parameter lainnya. Dalam hal ini, koordinat x diberikan oleh ekspresi cos(t)*cos(s)*d, koordinat y diberikan oleh ekspresi sin(t)*cos(s)*d, dan koordinat z diberikan oleh ekspresi sin(s)*d. Parameter "hue=1" digunakan untuk mengatur warna pada plot berdasarkan nilai fungsinya. Parameter "light=[1,0,1]" digunakan untuk mengatur pencahayaan pada plot. Parameter "frame=0" digunakan untuk menghilangkan frame pada plot. Parameter "zoom=5" digunakan untuk mengatur level zoom pada plot. **Grafik Tiga Dimensi yang**

---

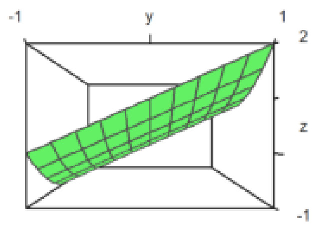
* Bersifat Interaktif dan animasi grafik 3D

Membuat gambar grafik tiga dimensi (3D) yang bersifat interaktif dan animasi grafik 3D adalah proses menciptakan visualisasi tiga dimensi yang memungkinkan pengguna berinteraksi dengan objek-objek 3D. Interaktivitas dalam gambar 3D memungkinkan pengguna untuk melakukan tindakan seperti mengubah sudut pandang, memindahkan objek, atau berinteraksi dengan elemen-elemen dalam adegan 3D. Animasi grafik 3D dapat mencakup pergerakan, tetapi juga dapat berarti perubahan dalam tampilan atau atribut objek tanpa pergerakan fisik yang mencolok.
CONTOH GAMBAR

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```

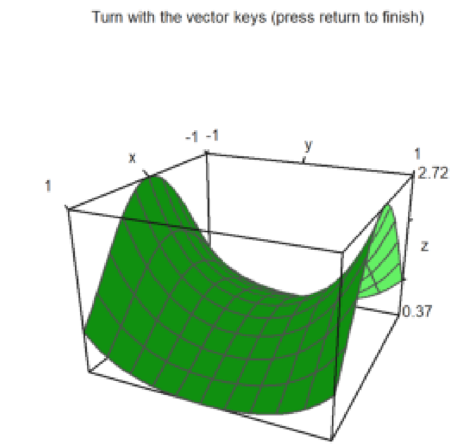



```
>function testplot () := plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0); ...
>rotate("testplot"); testplot():
```



Hilangkan command angle untuk bisa merotasikan grafik,dan height = 0 untuk membuat posisi sejajar dengan mata jadi tidak mempengaruhi pergerakan hanya berbeda sudut pandang saja

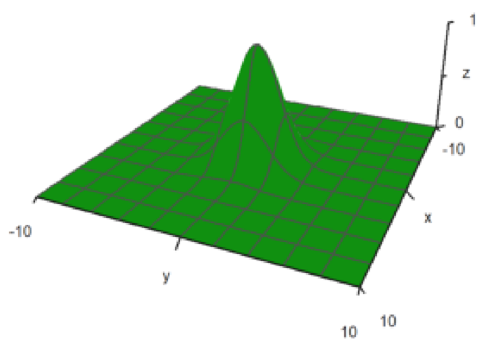
```
>plot3d("exp(-x^2+y^2)",>user, ...
>  title="Turn with the vector keys (press return to finish)"):
```




```
>plot3d("exp(x^2+y^2)",>user, ...
>title="Coba gerakan)")
```

Interaksi pengguna dimungkinkan dengan parameter. Pengguna dapat menekan tombol berikut.
1. kiri, kanan, atas, bawah: memutar sudut pandang
2. +,-: memperbesar atau memperkecil
3. a: menghasilkan anaglyph (lihat di bawah)
4. l: beralih memutar sumber cahaya (lihat di bawah)
5. spasi: disetel ulang ke default
6. enter: akhiri interaksi

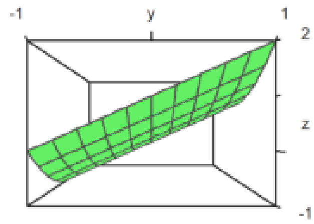
```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Parameter "r=10" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 10.
Parameter "n=80" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.
Parameter "fscale=4" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 4.
Parameter "scale=1.2" menunjukkan faktor skala pada plot. Semakin besar nilai scale, semakin besar ukuran

plot yang dihasilkan.
Parameter "frame=3" menunjukkan jenis frame yang digunakan pada plot. Dalam hal ini, jenis frame yang digunakan adalah frame kotak dengan sumbu x, y, dan z yang ditampilkan.

```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Tampilan dapat diubah dengan berbagai cara.
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1, angle=-20?, height=20?, ...
>  center=[0.4,0,0], zoom=5):
```

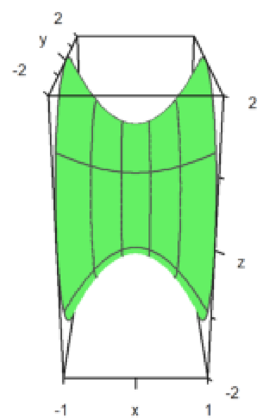
```
Closing bracket missing in function call!
Error in:
plot3d("x^4+y^2",a=0,b=1,c=-1,d=1, angle=-20?, height=20?,   c ...
                                            ^
```

Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.
Parameter center digunakan untuk memindahkan pusat plot ke lokasi tertentu dalam ruang. Dalam hal ini, pusat plot diatur ke titik (0.4, 0, 0) dalam ruang tiga dimensi. Parameter center berguna ketika kita ingin mengubah sudut pandang plot atau ketika kita ingin menyelaraskan plot dengan objek lain dalam scene. Dengan menentukan pusat plot, kita dapat mengontrol posisi kamera dan arah tampilan 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 diskalakan ke arah x dan y.
frame: jenis bingkai (default 1).

```
>function testplot () := plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50
>center=[0,0,-2],frame=3); ...
>rotate("testplot"); testplot():
```

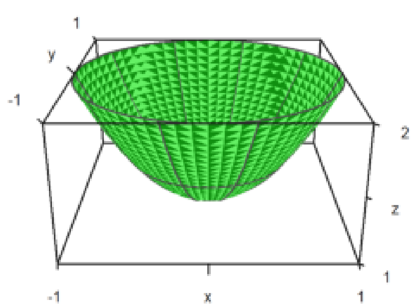
```
Closing bracket missing in function call!
testplot:
    useglobal; return plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale ...
Try "trace errors" to inspect local variables after errors.
rotate:
    f$(args());
```

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```

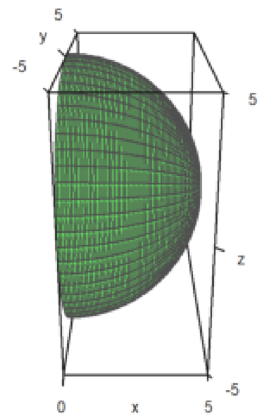


Penjelasan:
Secara umum, parameter "a" dan "b" digunakan untuk menentukan rentang nilai variabel independen dalam suatu fungsi. Dalam kasus ini, "a=-1" dan "b=1" menunjukkan bahwa fungsi tersebut akan diplot pada interval [-1, 1]. Parameter "rotate=true" menunjukkan bahwa grafik akan diputar untuk memberikan tampilan bentuk tiga dimensi yang lebih baik. Parameter "grid=5" menunjukkan bahwa grid dengan jarak 5 unit akan ditampilkan pada grafik.
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=2,grid=5):
```



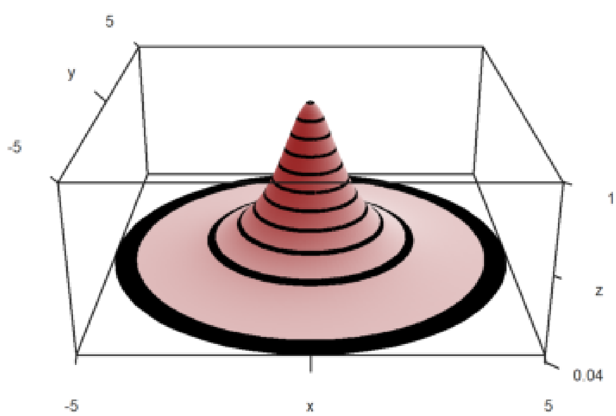
```
>function testplot () := plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1); ...
>rotate("testplot"); testplot():
```




```
>function testplot () := plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,height=20?, ...
>center=[0.4,0,0],zoom=5); ...
>rotate("testplot"); testplot():
```

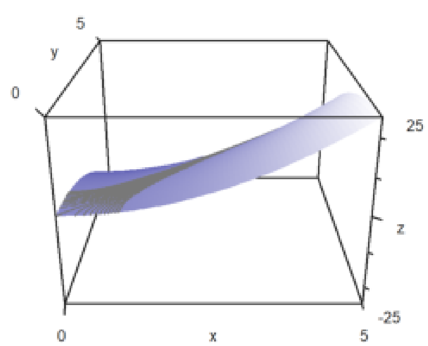
```
Closing bracket missing in function call!
testplot:
    useglobal; return plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,height=20 ...
Try "trace errors" to inspect local variables after errors.
rotate:
    f$(args());
```

```
>function testplot () := plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=red); ...
>rotate("testplot"); testplot():
```



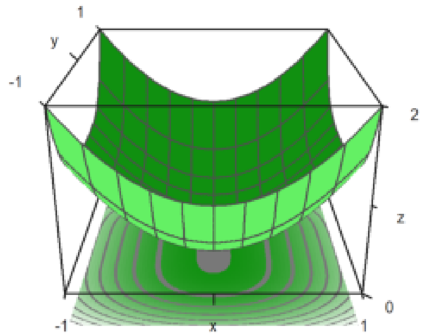
Parameter "r=5" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 5.
Parameter ">polar" menunjukkan bahwa plot yang dibuat adalah plot polar tiga dimensi. Plot polar adalah plot yang dibuat dengan menggunakan koordinat polar, yaitu koordinat yang terdiri dari jarak dan sudut.
Parameter "fscale=2" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 2.
Parameter ">hue" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.
Parameter "n=100" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.
Parameter "zoom=4" menunjukkan level zoom pada plot.
Parameter ">contour" menunjukkan bahwa garis kontur akan ditampilkan pada plot.
Parameter "color=blue" menunjukkan warna garis kontur pada plot. Dalam hal ini, warna yang digunakan adalah biru.
Untuk plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis level dan satu warna atau warna spektral. Euler dapat menggambar ketinggian fungsi pada sebuah plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.
-hue: Mengaktifkan bayangan cahaya, bukan kabel.
-contour: Membuat plot garis kontur otomatis pada plot.
-level=... (atau level): Vektor nilai garis kontur.

```
>function testplot () := plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=blue); ...
>rotate("testplot"); testplot():
```



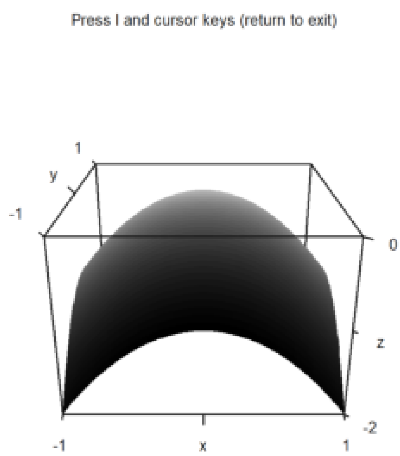
Parameter "level=-1:0.1:1" menunjukkan rentang nilai fungsinya yang akan ditampilkan pada plot. Dalam hal ini, rentang nilai fungsinya adalah dari -1 hingga 1 dengan interval 0.1.

```
>function testplot () := plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2); ...
>rotate("testplot"); testplot():
```



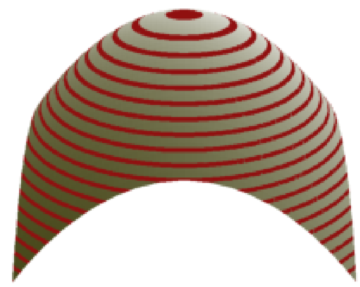
Parameter ">cp" menunjukkan bahwa titik kontrol akan ditambahkan pada plot. Titik kontrol digunakan untuk menentukan bentuk dan posisi plot tiga dimensi.
Parameter "cpcolor=green" menunjukkan warna titik kontrol yang akan digunakan. Dalam hal ini, warna yang digunakan adalah hijau.
Parameter "cpdelta=0.2" menunjukkan jarak antara titik kontrol. Semakin kecil nilai cpdelta, semakin banyak titik kontrol yang akan ditambahkan pada plot.

```
>plot3d("-x^2-y^2", ...
>hue=true,light=[0,1,1],amb=0,user=true, ...
> title="Press l and cursor keys (return to exit)"):
```



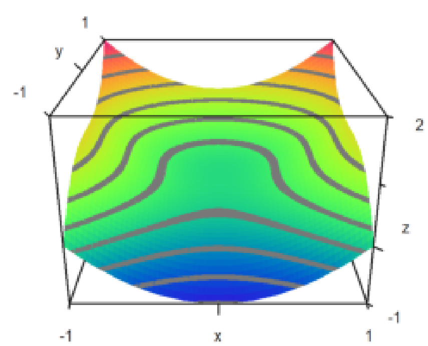
parameter "hue=true" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.
Parameter "light=light=[0,1,1] menunjukkan intensitas cahaya pada plot. Nilai light=[0,1,1] menunjukkan bahwa cahaya datang dari arah positif y dan z.
Parameter "amb=0" menunjukkan intensitas cahaya ambient pada plot. Nilai 0 menunjukkan bahwa tidak ada cahaya ambient yang digunakan.

```
>function testplot () := 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); ...
>rotate("testplot"); testplot():
```

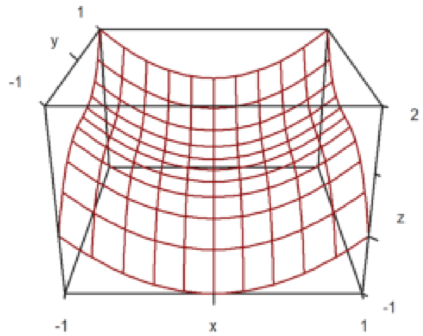


Parameter "frame=false" digunakan untuk menghilangkan frame pada plot tiga dimensi. Parameter "color=rgb(0.2,0.2,0)" menunjukkan warna dasar plot. Dalam hal ini, warna yang digunakan adalah hitam dengan nilai RGB (0.2, 0.2, 0). Parameter "dl=0.01" menunjukkan jarak antara titik-titik pada plot. Semakin kecil nilai dl, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat. Namun, semakin kecil nilai dl, semakin lama waktu yang dibutuhkan untuk membuat plot.

```
>function testplot () := plot3d("x^2+y^3",>contour,>spectral); ...
>rotate("testplot"); testplot():
```



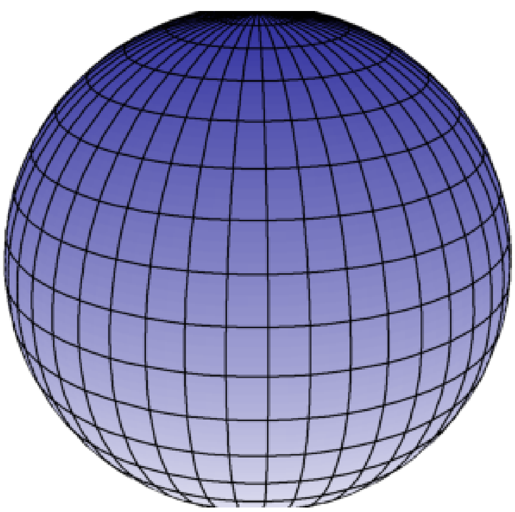

```
>function testplot () := plot3d("x^2+y^3", >transparent, grid=10, wirecolor=red); ...
>rotate("testplot"); testplot():
```



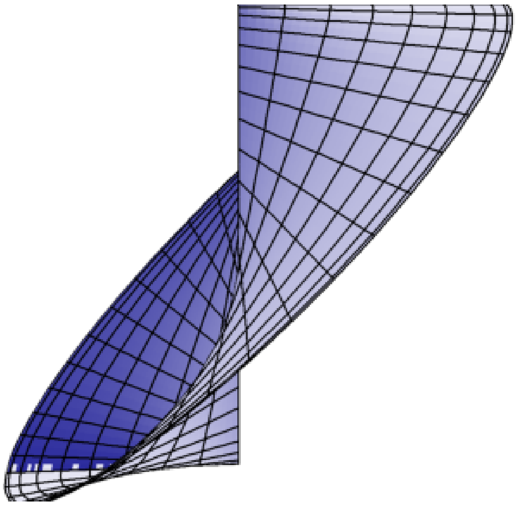
Fungsi Parametrik 3D

Fungsi parametrik merupakan jenis fungsi matematika yang menggambarkan hubungan antara dua atau lebih variabel, dimana masing-masing koordinat (x, y, z...) dinyatakan sebagai fungsi lain dari beberapa parameter. Fungsi parametrik dapat digunakan untuk menggambar kurva, lintasan, atau hubungan antara berbagai variabel yang bergantung pada parameter-parameter tertentu.
Sebagai contoh :

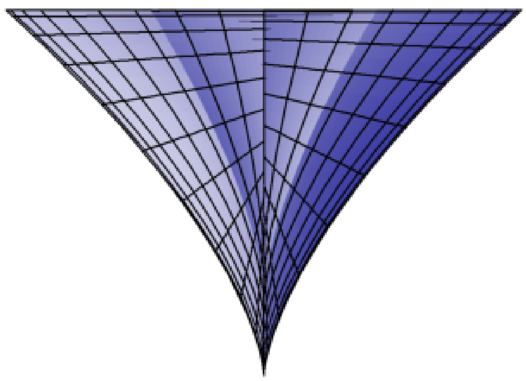
```
>plot3d("cos(x)*cos(y)","sin(x)*cos(y)","sin(y)", a=0,b=2*pi,c=pi/2,d=-pi/2,...
>>hue,color=blue,light=[0,1,3],<frame,...
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



```
>plot3d("cos(x)*cos(y)","sin(x)*cos(y)","cos(x)", a=0,b=2*pi,c=pi/2,d=-pi/2,...
>>hue,color=blue,light=[0,1,3],<frame,...
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



```
>plot3d("cos(x)^3*sin(y)","sin(x)^2*sin(y)","cos(x)^2", a=0,b=2*pi,c=pi/2,d=-pi/2,...
>>hue,color=blue,light=[0,1,5],<frame,...
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



8 Menggambar Fungsi Implisit Implisit

Fungsi implisit (implicit function) adalah fungsi yang memuat lebih dari satu variabel, berjenis variabel bebas dan variabel terikat yang berada dalam satu ruas sehingga tidak bisa dipisahkan pada ruas yang berbeda.

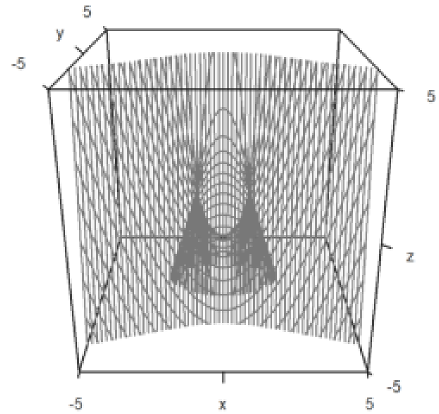
$$F(x, y, z) = 0$$

(1 persamaan dan 3 variabel), terdiri dari 2 variabel bebas dan 1 terikat
Misalnya,

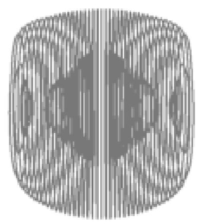
$$F(x, y, z) = x^2 + y^2 + z^2 = 1$$

adalah persamaan implisit yang menggambarkan bola dengan jari-jari 1 dan pusat di (0,0,0).

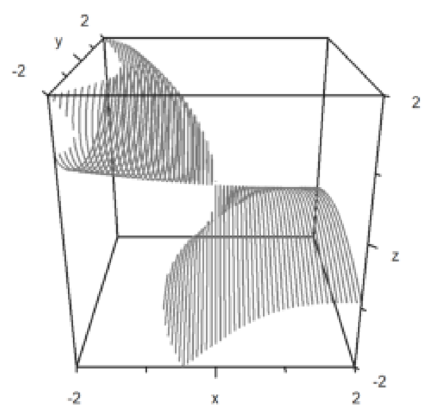
```
>plot3d("x^2+y^3+z*y-1", r=5, implicit=3):
```




```
>c=1; d=1;
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit, r=2, zoom=2.5):
```



Selain plot kontur yang sudah di jelaskan sebelumnya, pada EMT juga ada plot umplisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel.
Solusi dari

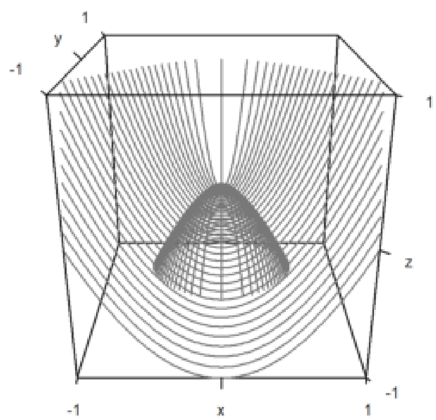
$$f(x, y, z) = 0$$

dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.
- implisit = 1: potong sejajar dengan bidang-y-z
- implicit = 2: memotong sejajar dengan bidang x-z
- implicit=4: memotong sejajar dengan bidang x-y

Ambil contoh dari persamaan latex pada fungsi implisit tadi dan tambahkan nilai-nilai ini, sehingga kita dapat memplot persamaan ini

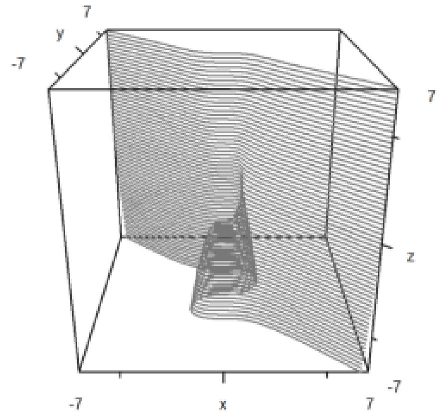
$$M = (x, y, z) : x^2 + y^3 + zy = 1$$

```
>plot3d("x^2+y^3+z*y", r=1, implicit=2):
```

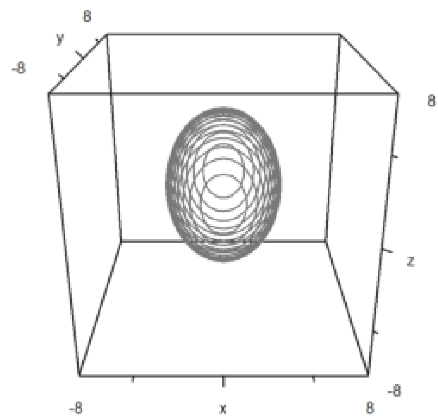


Contoh fungsi implisit yang lainnya

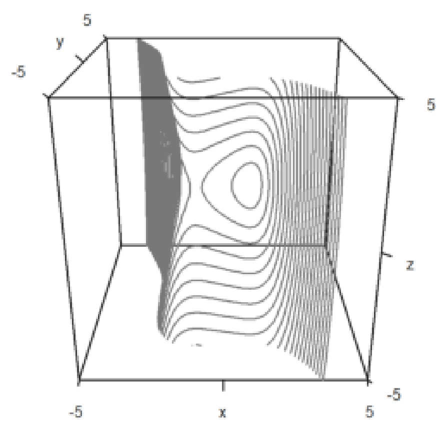
```
>plot3d("x^3+y^3+z*y-1",r=7,implicit=4):
```



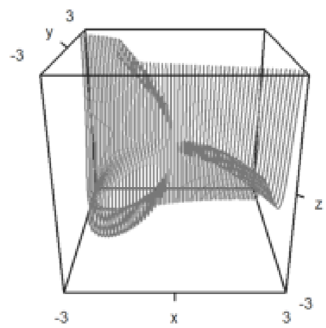

```
>plot3d("2*x^2 + 3*y^2 + z^2 - 25",r=8,implicit=2):
```



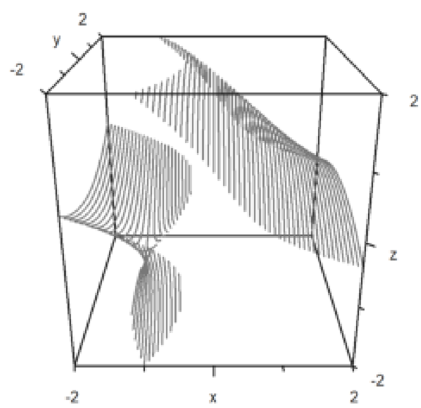

```
>plot3d("x^5 + 5*y^3 + 3*z^2 - 5*x - 7*y - 5*z + 10",r=5,implicit=2):
```



```
>plot3d("x^3+y^5+5*x*z+z^3",>implicit,r=3,zoom=2):
```




```
>plot3d("x^2+y^2+4*x*z+z^3-5",>implicit,r=2,zoom=2.5):
```



Fungsi Implisit Menggunakan Povray

Povray dapat memplot himpunan di mana f(x,y,z)=0, seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.
Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)=d$$

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

```
>povstart(angle=70°,height=50°, zoom=4);
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(blue)));
>writeAxes();
>povend();
```

```
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);
```

BAB 4

KB PEKAN 6-7: MENGGUNAKAN EMT UNTUK KALKULUS

[a4paper,10pt]article eumat

BAB 6. BARISAN DAN DERET

MATERI BARISAN DAN DERET PADA EMT Dibuat Oleh :
NAMA : AKHMAD RIZKI KHAMID
KELAS : MATEMATIKA E
NIM : 22305141008

CAKUPAN MATERI MELIPUTI DIANTARANYA :

```
1. Mendefinisikan Barisan dan Deret pada EMT
2. BARISAN
   A. Definisi Barisan
   B. Limit Barisan
   C. Contoh Soal Barisan & Visualisasi
3. DERET
   3.1 DERET BERHINGGA
       A. Definisi
       B. Rumus deret berhingga
       C. Contoh Soal & Visualisasi
   3.2 DERET TAK HINGGA
       A. Definisi
       B. Rumus deret tak hingga
       C. Contoh Soal & Visualisasi
4. KEKONVERGENAN BARISAN DAN DERET
5. APLIKASI BARISAN DAN DERET
```

```
1. Mendefinisikan Barisan dan Deret pada EMT
```

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:
- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (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;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

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.

BARISAN —

```
A. Definisi Barisan adalah sebuah daftar bilangan yang mengurut
```

dari kiri ke kanan. Setiap urutan bilangannya juga memiliki karakteristik atau pola tertentu. Setiap bilangan yang ada pada barisan merupakan suku dalam barisan itu sendiri.

```
B. Limit Barisan
```

limit barisan adalah nilai yang didekati oleh suku-suku barisan ketika nomor urut suku-sukunya semakin membesar. Limit barisan seringkali dilambangkan dengan {\displaystyle \lim } (yaitu, {\textstyle \lim _{n\to \infty }a_{n}}.Jika suatu barisan mempunyai limit, barisan itu disebut konvergen. Barisan yang tidak konvergen disebut divergen.Limit barisan dikatakan sebagai gagasan landasan seluruh analisis matematika

```
C. Contoh Soal Barisan & Visualisasi
```

1.Barisan aritmatika adalah sebagai berikut:
Barisan aritmatika adalah urutan atau rangkaian bilangan di mana setiap angka atau elemen dalam barisan ini memiliki selisih yang tetap dengan angka sebelumnya.

```
Selisih ini disebut beda aritmatika, dan biasanya digunakan untuk
```

menghasilkan anggota-anggota berikutnya dalam barisan. Dalam barisan aritmatika, setiap angka (kecuali angka pertama) ditemukan dengan menambahkan beda aritmatika ke angka sebelumnya.

$$a_n = a_1 + (n-1)d$$

$$a_n$$

adalah anggota ke-n dalam barisan aritmatika.

$$a_1$$

adalah anggota pertama dalam barisan aritmatika.

$$n$$

adalah posisi anggota yang ingin ditentukan.

$$d$$

adalah beda aritmatika (perbedaan yang tetap antara setiap anggota).
Contoh Soal sederhana Barisan Aritmatika:

```
>1:20
```

```
[1,  2,  3,  4,  5,  6,  7,  8,  9,  10,  11,  12,  13,  14,  15,  16,
17,  18,  19,  20]
```

```
>10:70
```

```
[10,  11,  12,  13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,
24,  25,  26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,
38,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,
66,  67,  68,  69,  70]
```

```
>2:200
```

```
[2,  3,  4,  5,  6,  7,  8,  9,  10,  11,  12,  13,  14,  15,  16,  17,
18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,
32,  33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,  44,  45,
46,  47,  48,  49,  50,  51,  52,  53,  54,  55,  56,  57,  58,  59,
60,  61,  62,  63,  64,  65,  66,  67,  68,  69,  70,  71,  72,  73,
74,  75,  76,  77,  78,  79,  80,  81,  82,  83,  84,  85,  86,  87,
88,  89,  90,  91,  92,  93,  94,  95,  96,  97,  98,  99,  100,  101,
102,  103,  104,  105,  106,  107,  108,  109,  110,  111,  112,  113,
114,  115,  116,  117,  118,  119,  120,  121,  122,  123,  124,  125,
126,  127,  128,  129,  130,  131,  132,  133,  134,  135,  136,  137,
138,  139,  140,  141,  142,  143,  144,  145,  146,  147,  148,  149,
150,  151,  152,  153,  154,  155,  156,  157,  158,  159,  160,  161,
162,  163,  164,  165,  166,  167,  168,  169,  170,  171,  172,  173,
174,  175,  176,  177,  178,  179,  180,  181,  182,  183,  184,  185,
186,  187,  188,  189,  190,  191,  192,  193,  194,  195,  196,  197,
198,  199,  200]
```

- Contoh Sederhana Barisan Aritmatika dengan Selisih tertentu dan memiliki Batasan

```
>1:4:21
```

```
[1,  5,  9,  13,  17,  21]
```

```
>2:7:51
```

```
[2,  9,  16,  23,  30,  37,  44,  51]
```

```
>5:10:100
```

```
[5,  15,  25,  35,  45,  55,  65,  75,  85,  95]
```

- Contoh Soal Barisan Aritmatika

```
Sebuah barisan aritmatika memiliki anggota pertama adalah 2 dengan
```

beda 5. Hitunglah anggota (suku) ke-50 dari barisan tersebut!

```
>function f(n):=2+(n-1)*5
>f(50)
```

```
247
```

- Contoh Soal cerita sederhana Barisan aritmatika

```
Akhmad ingin menabung setiap bulannya untuk persiapan liburan dengan
```

temannya. Pada bulan pertama, dia menyimpan Rp 300.000,00. Setiap bulan berikutnya, dia ingin menambahkan Rp 100.000,00 lebih banyak daripada bulan sebelumnya. Berapa total uang yang akan dia simpan setelah 12 bulan?

```
>function f(n):=300000+(n-1)*100000
>f(12)
```

```
1400000
```

2.Barisan Geometri
Barisan geometri adalah barisan bilangan dimana dua suku yang berurutan memiliki perbandingan yang sama,urutan bilangan di mana setiap angka atau elemen dalam barisan ini ditemukan dengan mengalikan angka sebelumnya dengan suatu bilangan tetap yang disebut rasio geometri.

```
Rasio geometri ini digunakan untuk menghasilkan anggota-anggota
```

berikutnya dalam barisan. Dalam barisan geometri, setiap angka (kecuali angka pertama) ditemukan dengan mengalikan angka sebelumnya dengan rasio geometri.
^Rumus^

```
Rumus umum untuk barisan geometri adalah
```

$$a_n = a_1 r^{(n-1)}$$

Dimana:

$$a_n$$

adalah anggota ke-n dalam barisan geometri.

$$a_1$$

adalah anggota pertama dalam barisan geometri.

$$r$$

adalah rasio geometri (perbandingan tetap antara setiap anggota dalam barisan).

$$n$$

adalah posisi anggota yang ingin ditemukan.

- Contoh Sederhana Barisan Geometri

Tentukan nilai suku barisan geometri dari suku ke-1 sampai ke-10, jika diketahui suku pertama adalah 4, dengan rasio 2!

```
>a1:=4
```

```
4
```

```
>r:=2
```

```
2
```

```
>n:=(1:10)
```

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

```
>function f(n):= a1*r^(n-1)
>f(n)
```

```
[4,  8,  16,  32,  64,  128,  256,  512,  1024,  2048]
```

- Contoh Soal cerita sederhana barisan Geometri

1. Siti memiliki satu kotak berisi permen. Setiap hari, dia membagi

permen di dalam kotak itu menjadi dua, kemudian menambahkan satu permen lagi ke dalam kotak. Pada hari pertama, dia memiliki 4 permen dalam kotak.Berapa banyak permen yang akan ada dalam kotak Siti pada hari ke-5?

```
>a1:=4
```

```
4
```

```
>r:=2
```

```
2
```

```
>n:=(5)
```

```
5
```

```
>function f(n):=a1*r^(n-1)
>f(5)
```

```
64
```

    2. Bagas adalah seorang kolektor barang antik. Setiap bulan, dia

menemukan barang antik ke koleksinya. Awalnya, koleksinya hanya memiliki satu barang antik langka yang dia beli seharga Rp 300.000,00.Bagas memutuskan untuk terus menambahkan barang antik setiap bulan dengan rasio pertumbuhan harga sebesar 10 %. Dia ingin mengetahui berapa harga barang antik yang akan menjadi Koleksi ke-6nya.

```
>a1:=300000
```

```
300000
```

```
>r:=(110/100)
```

```
1.1
```

```
>function f(n):= a1*r^(n-1)
>f(6)
```

```
483153
```

ITERASI

A. DEFINISI

Di dalam ilmu komputer Iterasi diartikan sebagai sifat tertentu

dari algoritma atau program komputer dimana suatu urutan atau lebih dari langkah algoritmik dilakukan di loop program (program pengulangan)

Sedangkan dalam ilmu matematika, iterasi dapat diartikan sebagai

proses atau metode yang digunakan secara berulang-ulang (pengulangan) dalam menyelesaikan permasalahan matematik.
B. CONTOH SOAL

1. Pertumbuhan penduduk dari nilai awal 1000 dengan laju

pertambahan 5%, selama 5 tahun.

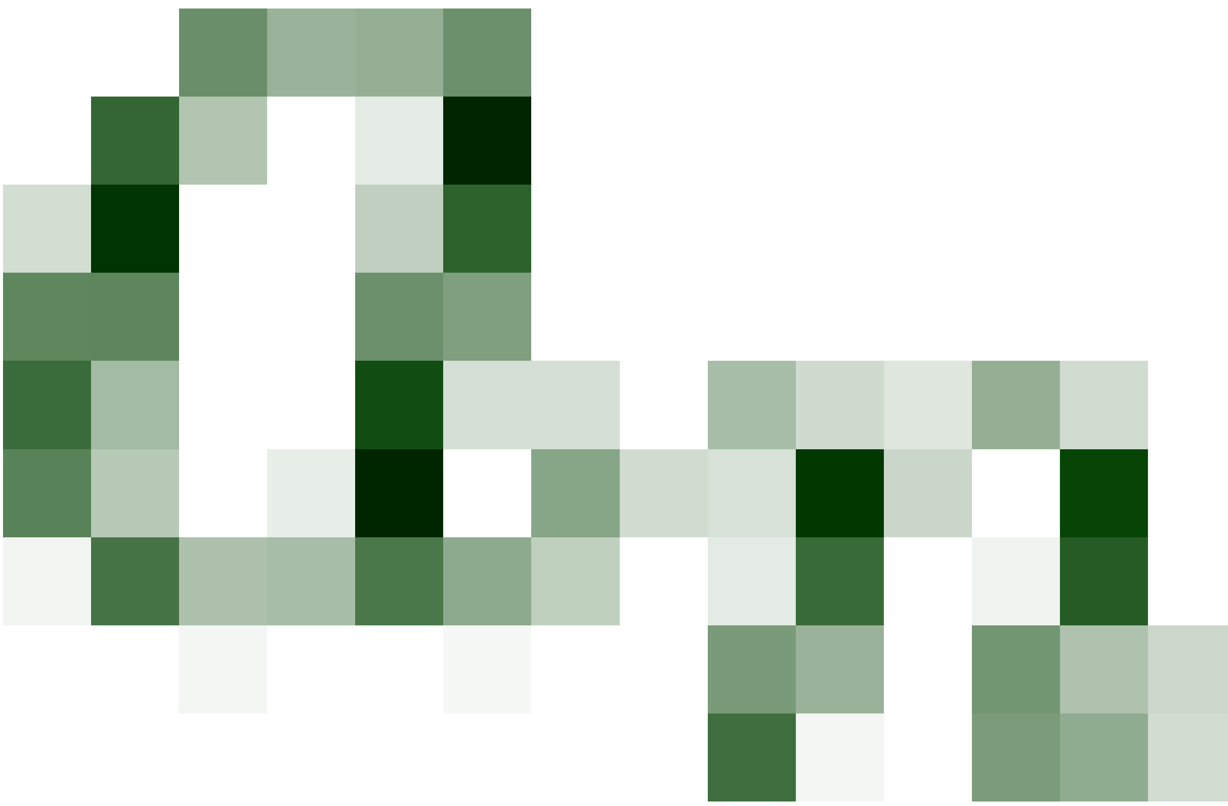
```
>r=1.05; iterate("x*r",1000,n=5)
```

```
[1000,  1050,  1102.5,  1157.63,  1215.51,  1276.28]
```

2. Bahaya menabung di bank pada masa sekarang! Dengan bunga

tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan,
jika memiliki tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```



sintax untuk mencari minimal tabungan setiap bulannya agar tidak habis diambil oleh bank:

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

$$\left[A=\frac{5\,a}{4\,r}\right]$$

$$[A=2500.0]$$

Masih terkait dengan contoh soal ke-2, kemudian akan ditampilkan fungsi untuk menghitung saldo tabungan dan kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=7)
```

```
[1000,  994,  987.98,  981.93,  975.86,  969.76,  963.64,  957.49]
```

```
>iterate({{"saldo",0.005,10}},2000,n=7)
```

```
[2000,  1998,  1995.99,  1993.97,  1991.95,  1989.92,  1987.88,
1985.83]
```

```
>iterate({{"saldo",0.005,10}},21000,n=7)
```

```
[21000,  21074,  21148.3,  21222.9,  21297.8,  21373,  21448.5,
21524.3]
```

```
>iterate({{"saldo",0.005,10}},23000,n=7)
```

```
[23000,  23082,  23164.3,  23247,  23330,  23413.3,  23497,  23580.9]
```

```
>iterate({{"saldo",0.005,10}},25000,n=7)
```

```
[25000,  25090,  25180.4,  25271.1,  25362.2,  25453.6,  25545.4,
25637.6]
```

BARISAN FIBONACCI

A. DEFINISI

```
Barisan Fibonacci adalah susunan angka dengan nilai angka
```

berikutnya diperoleh dari hasil menambahkan kedua angka sebelumnya secara berturut-turut.Sederhananya pola yang suku bilangannya merupakan penjumlahan dua suku sebelumnya

```
Barisan Fibonacci termasuk barisan yang kompleks yang bisa
```

menggunakan fungsi "sequence()". Fungsi "sequence()" digunakan untuk menghitung nilai-nilai x[n] dari semua nilai sebelumnya, x[1],...,x[n-1] yang diketahui.
B. CONTOH

```
- Barisan Fibonacci
```

$$x_n = x_n - 1 + x_n - 2, x_1 = 1, x_2 = 1$$

```
pengaplikasian dalam EMT
```

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
```

```
[1,  1,  2,  3,  5,  8,  13,  21,  34,  55,  89,  144,  233,  377,  610]
```

$$x_n = x_n - 1 + x_n - 1, x_1 = 7, x_2 = 14$$

```
pengaplikasian dalam EMT
```

```
>sequence("x[n-1]+x[n-1]",[7,14],15)
```

```
[7,  14,  28,  56,  112,  224,  448,  896,  1792,  3584,  7168,  14336,
28672,  57344,  114688]
```

C. SIFAT MENARIK BARISAN FIBONACCI

```
Sifat menariknya yaitu akar pangkat ke-n suku ke-n akan konvergen
```

ke pecahan emas.
Contohnya yaitu :

```
>$(1+sqrt(7))/2=float((1+sqrt(7))/2)
```

$$\frac{\sqrt{7}+1}{2} = 1.822875655532295$$

```
>$(1+sqrt(6))/2=float((1+sqrt(6))/2)
```

$$\frac{\sqrt{6}+1}{2} = 1.724744871391589$$

```
>$(15+sqrt(22))/2=float((15+sqrt(22))/2)
```

$$\frac{\sqrt{22}+15}{2} = 9.845207879911715$$

Untuk plot 2d nya:

```
>plot2d(sequence("x[n-1]+x[n-2]",[1,1],70)^(1/(1:70))):
```

$$\frac{\sqrt{22}+15}{2}$$

```
Dalam Baris Fibonacci terdapat yang namanya Rekursi. Rekursi
```

berfungsi untuk mendefinisikan urutan atau deret matematika dengan cara yang berulang ata belaku untuk nilai yang lebih kecil. Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya.
Contohnya elemen ke-n merupakan jumlah (n-1) elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah 2^(n-2).

```
>sequence("sum(x)",1,15)
```

```
[1,  1,  2,  4,  8,  16,  32,  64,  128,  256,  512,  1024,  2048,
4096,  8192]
```

```
>sequence("sum(x)",7,12)
```

```
[7,  7,  14,  28,  56,  112,  224,  448,  896,  1792,  3584,  7168]
```

```
>sequence("sum(x)",-3,8)
```

```
[-3,  -3,  -6,  -12,  -24,  -48,  -96,  -192]
```

```
>sequence("sum(x)",-8,15)
```

```
 [-8,  -8,  -16,  -32,  -64,  -128,  -256,  -512,  -1024,  -2048,
 -4096,  -8192,  -16384,  -32768,  -65536]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.
Pada contoh berikut, digunakan iterasi

$$x_n = A \cdot x_{n-1},$$

dengan A suatu matriks 2x2, dan setiap x[n] merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],4)
```

```
              1             2             5            13
              1             3             8            21
```

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak log_2(n).

$$x_n = A.x_{n-1} = A^2.x_{n-2} = A^3.x_{n-3} = ... = A^{n-1}.x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,4)
```

```
              1             5            13            34
              1             8            21            55
```

SPIRAL TEODORUS

Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

```
>function g(n) := 1+I/sqrt(n)
```

Rekursinya dapat dijalankan sebanyak n untuk menghasilkan barisan n bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya
CONTOH :

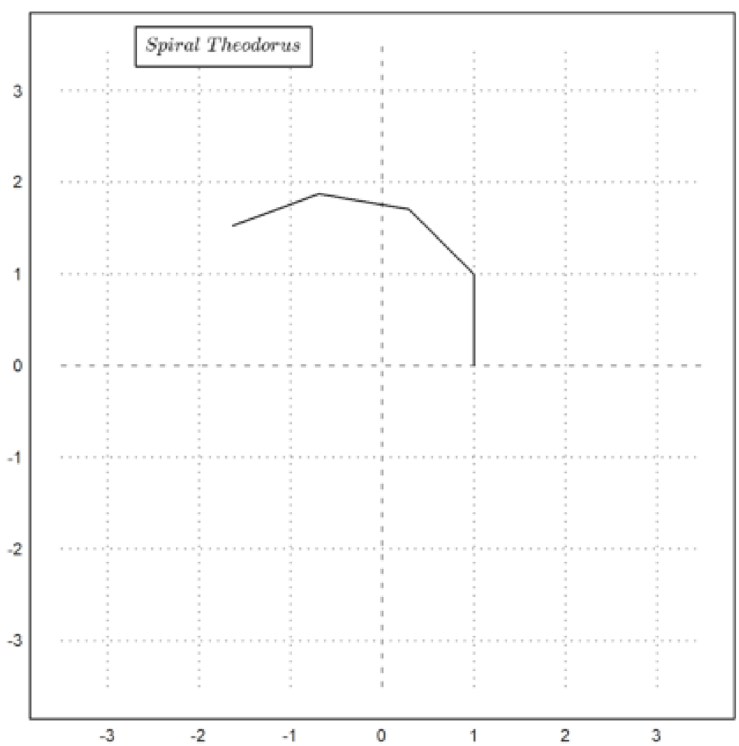
```
1. Untuk n=5
```

```
>x=sequence("g(n-1)*x[n-1]",1,5); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4)
```

$$x_n = \left(1 + \right.$$

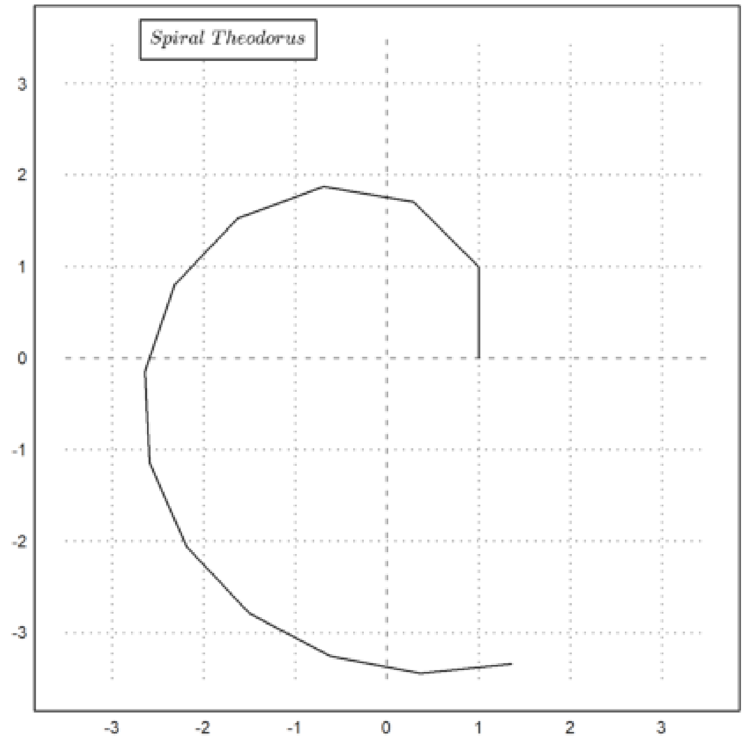
2. Untuk n=13

```
>x=sequence("g(n-1)*x[n-1]",1,13); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4
```



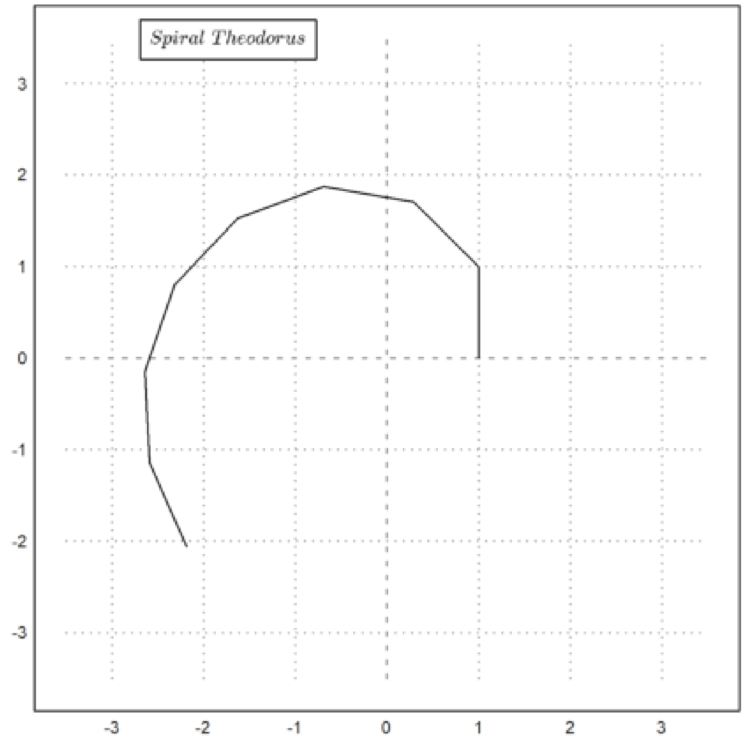
3. Untuk n=26

```
>x=sequence("g(n-1)*x[n-1]",1,9); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4)
```



Selanjutnya titik-titik kompleks dari barisan yang sudah ditampilkan pada plot akan dihubungan ke titik 0 menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```

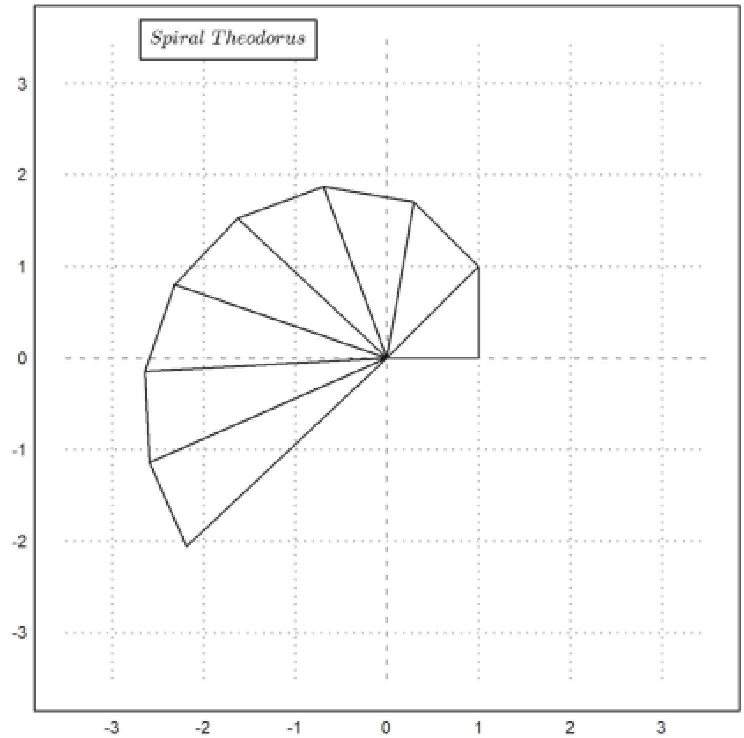



```
>function gstep (v) ...
```

```
  w=[-v[2];v[1]];
  return v+w/norm(w);
  endfunction
```

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memmerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
>x=iterate("gstep",[1;0],7); plot2d(x[1],x[2],r=3.5,>points):
```



Deret — Deret adalah rangkaian atau barisan suku-suku bilangan

yang dijumlahkan satu per satu sesuai dengan suatu aturan atau pola tertentu. Deret dapat berupa rangkaian bilangan riil, bilangan bulat, bilangan kompleks, atau bahkan fungsi matematika.
Secara umum, suku-suku dalam deret dinotasikan sebagai

$$a_1, a_2, a_3, ...$$

Deret terdiri dari deret berhingga dan deret tak hingga

Deret Berhingga

Deret Berhingga adalah konsep dalam matematika yang mengacu pada penjumlahan suku-suku berhingga dari suatu deret numerik. Deret ini terdiri dari suku-suku yang membentuk rangkaian berhingga yang terbatas. Definisi formalnya adalah sebagai berikut.
Sebuah deret berhingga adalah rangkaian berhingga suku

$$a_1, a_2, a_3, ..., a_n$$

yang dijumlahkan untuk membentuk jumlah yang berhingga. Deret ini biasanya dinyatakan sebagai

$$S_n = a_1 + a_2 + a_3 + ... + a_n$$

Di mana S adalah jumlah dari deret berhingga tersebut.

```
>$'sum(k, k, 1, n)
```

$$\sum_{k=1}^{n} k$$

Perintah tersebut digunakan untuk menghitung jumlah dari deret berhingga

$$\sum_{k=n}^{n} k$$

yang dimulai dari k=n hingga n yang ditentukan, dan kemudian mencoba untuk menyederhanakan hasilnya secara simbolik dengan menggunakan perintah ev dan opsi simpsum=true.
sum(a, k, n, n): Ini adalah ekspresi yang digunakan untuk menghitung jumlah deret berhingga

$$\sum_{k=n}^{n} k$$

dengan k sebagai variabel indeks, dimulai dari k=n hingga n yang ditentukan. Ini adalah notasi umum untuk menggambarkan deret berhingga yang dimulai dari k=n hingga n yang ditentukan.
=: Ini adalah operator yang digunakan untuk menetapkan hasil perhitungan di sebelah kiri kepada ekspresi di sebelah kanan. Dalam hal ini, pernyataan ini mengatakan bahwa hasil dari perhitungan deret sebelah kiri harus sama dengan hasil dari perhitungan deret sebelah kanan.
factor(...) Ini adalah fungsi yang digunakan untuk memfaktorkan ekspresi matematika. Dalam konteks ini, hasil dari jumlah deret difaktorkan.
ev(...) Ini adalah perintah dalam bahasa yang digunakan untuk mengevaluasi (menghitung) ekspresi matematika.
simpsum=true Ini adalah opsi yang digunakan bersamaan dengan perintah ev untuk memberitahu bahwa Anda ingin mencoba menyederhanakan hasil perhitungan deret secara simbolik jika memungkinkan. Dengan kata lain, opsi ini menginstruksikan untuk melakukan penyederhanaan pada hasil deret.
Macam-Macam Deret Berhingga
1. Deret Aritmatika
Deret dengan suku-suku yang memiliki beda tetap antara setiap dua suku berturut-turut.
Rumus deret aritmatika

$$S_n = \frac{n}{2}(a + U_n)$$

Berikut ini contoh perintah yang digunakan dalam deret aritmatika untuk beda b=1

```
>$'sum(k, k, 1, 7) = factor(ev(sum(k, k, 1, 7),simpsum=true))// simpsum:menghitung deret s
```

$$\sum_{k=1}^{7} k = 28$$

Perintah tersebut menghitung jumlah deret berhingga

$$\sum_{k=1}^{7} k$$

Deret tersebut adalah deret geometri dengan suku awal k=1
dan beda b=1. Deret geometri ini memiliki susku-suku
1+2+3+...+7
yang dimulai dari k=1 hingga 7. Hasil dari perhitungan deret geometri tersebut adalah 28.

```
>$'sum(k, k, 5, 25) = factor(ev(sum(k, k, 5, 25),simpsum=true)) // simpsum:menghitung dere
```

$$\sum_{k=5}^{25}k=315$$

Perintah tersebut menghitung jumlah deret berhingga

$$\sum_{k=5}^{25}k$$

Deret tersebut adalah deret geometri dengan suku awal k=5
dan beda b=1. Deret geometri ini memiliki susku-suku
5+6+...+25
yang dimulai dari k=5 hingga 25. Hasil dari perhitungan deret geometri tersebut adalah 315.

```
>$'sum(a+(n-1)*b, n, 1, 7) = factor(ev(sum(4+(n-1)*8, n, 1, 7),simpsum=true))
```

$$\sum_{n=1}^{7}\left(b\,\left(n-1\right)+a\right)=196$$

Perintah tersebut menghitung jumlah deret berhingga sebanyak 7 suku

$$\sum_{k=4}^{n}k$$

Deret tersebut adalah deret geometri dengan suku awal k=4
dan beda b=8. Deret geometri ini memiliki 7 suku yang dimulai dari k=4 hingga suku ke-7 nya. Hasil dari perhitungan deret geometri tersebut adalah 196.

```
>$'sum(a+(n-1)*b, n, 1, 15) = factor(ev(sum(3+(n-1)*2, n, 1, 15),simpsum=true))
```

$$\sum_{n=1}^{15}\left(b\,\left(n-1\right)+a\right)=255$$

Perintah tersebut menghitung jumlah deret berhingga sebanyak 15 suku

$$\sum_{k=3}^{n}k$$

Deret tersebut adalah deret geometri dengan suku awal k=3
dan beda b=2. Deret geometri ini memiliki 15 suku yang dimulai dari k=3 hingga suku ke-15 nya. Hasil dari perhitungan deret geometri tersebut adalah 255.

2. Deret Geometri
Deret dengan suku-suku yang memiliki rasio tetap antara setiap dua suku berturut-turut.
Untuk r<1

$$S_n=\frac{a(1-r^n)}{1-r}$$

Untuk r>1

$$S_n=\frac{a(r_n-1)}{r-1}$$

```
>&powerdisp:false
```

false

```
>$'sum(a*r^(n-1), n, 1, 3) = factor(ev(sum(2*2^(n-1), n, 1, 3),simpsum=true))
```

$$a\sum_{n=1}^{3}r^{n-1}=14$$

Perintah tersebut menghitung jumlah deret berhingga sebanyak 3 suku

$$\sum_{n=3}^{n}(2(2^{(n-1)})$$

Deret tersebut adalah deret geometri dengan suku awal n=2
dan rasio r=2. Deret geometri ini memiliki 3 suku yang dimulai dari n=2 hingga suku ke-3 nya. Hasil dari perhitungan deret geometri tersebut adalah 14.

```
>$'sum(a*r^(n-1), n, 1, 2) = factor(ev(sum(8*10^(n-1), n, 1, 2),simpsum=true))
```

$$a\sum_{n=1}^{2}r^{n-1}=88$$

Perintah tersebut menghitung jumlah deret berhingga sebanyak 2 suku

$$\sum_{n=8}^{n}(8(10^{(n-1)})$$

Deret tersebut adalah deret geometri dengan suku awal n=8
dan rasio r=10. Deret geometri ini memiliki 2 suku yang dimulai dari n=2 hingga suku ke-2 nya. Hasil dari perhitungan deret geometri tersebut adalah 88.

```
>$'sum(a*r^(n-1), n, 1, 3) = factor(ev(sum(4*2^(n-1), n, 1, 3),simpsum=true))
```

$$a\sum_{n=1}^{3}r^{n-1}=28$$

Perintah tersebut menghitung jumlah deret berhingga sebanyak 3 suku

$$\sum_{n=1}^{3}(4(2^{(n-1)})$$

Deret tersebut adalah deret geometri dengan suku awal a=4
dan rasio r=2. Deret geometri ini memiliki 3 suku yang dimulai dari a=4 hingga suku ke-3 nya. Hasil dari perhitungan deret geometri tersebut adalah 28.

```
>$'sum(a*r^(n-1), n, 1, 10) = factor(ev(sum(1*2^(n-1), n, 1, 10),simpsum=true))
```

$$a\sum_{n=1}^{10} r^{n-1} = 1023$$

Perintah tersebut menghitung jumlah deret berhingga sebanyak 10 suku

$$\sum_{n=1}^{6}(1(2^{(n-1)})$$

Deret tersebut adalah deret geometri dengan suku awal a=1
dan rasio r=2. Deret geometri ini memiliki 10 suku yang dimulai dari a=1 hingga suku ke-10 nya. Hasil dari perhitungan deret geometri tersebut adalah 28.

4. Deret Eksponensial
Deret yang yang memiliki suku-suku yang didefinisikan oleh fungsi eksponensial

$$e^x$$

```
>$'sum(x^k, k, n, n) = factor(ev(sum(x^k, k, n, n),simpsum=true))
```

$$\sum_{k=n}^{n} x^k = x^n$$

Perintah tersebut adalah perintah yang digunakan untuk menghitung jumlah suku dalam deret matematika yang berbentuk eksponensial

$$x^k$$

dengan k= hingga k=n. Sehingga bentuk deretnya adalah

$$x^n + ... + x^n$$

```
>$'sum(x^k, k, 0, 3) = factor(ev(sum(x^k, k, 0, 3),simpsum=true))
```

$$\sum_{k=0}^{3} x^k = (x+1)\left(x^2+1\right)$$

bentuk lain

$$x^0 + x^1 + x^2 + x^3$$

```
>$'sum(x^k, k, 0, 3) = factor(ev(sum(2^k, k, 0, 3),simpsum=true))
```

$$\sum_{k=0}^{3} x^k = 15$$

Perintah tersebut adalah perintah yang digunakan untuk menghitung jumlah suku dalam deret matematika yang berbentuk eksponensial

$$2^0 + ... + 2^3$$

dengan k=0 hingga k=3. Sehingga hasil perhitungan dari suku-sukunya adalah 15

Deret Tak Hingga

Deret tak hingga adalah konsep dalam matematika yang mengacu pada penjumlahan suku-suku tak terhingga dari suatu deret numerik. Deret ini terdiri dari suku-suku yang membentuk rangkaian tak terhingga yang bisa berlanjut tanpa batas. Definisi formalnya adalah sebagai berikut.
Sebuah deret tak hingga adalah rangkaian tak terhingga suku

$$a_1, a_2, a_3, \ldots$$

yang dijumlahkan untuk membentuk jumlah tak terhingga. Deret ini biasanya dinyatakan sebagai

$$S_n = a_1 + a_2 + a_3 + \ldots$$

Di mana S adalah jumlah dari deret tak hingga tersebut.
Penting untuk diingat bahwa dalam matematika, untuk mengklasifikasikan deret tak hingga sebagai konvergen (mempunyai jumlah terhingga) atau divergen (mempunyai jumlah tak terhingga), kita perlu memeriksa perilaku suku-suku deret ini ketika jumlahnya mendekati tak terhingga (yaitu, ketika n mendekati tak terhingga). Terdapat berbagai metode dan kriteria yang digunakan untuk menentukan apakah deret tak hingga tertentu konvergen atau divergen.

```
>$'sum(k, k , n , inf)
```

$$\sum_{k=n}^{\infty} k$$

Perintah tersebut digunakan untuk menghitung jumlah dari deret tak hingga

$$\sum_{k=n}^{\infty} k$$

yang dimulai dari x=n hingga tak terhingga, dan kemudian mencoba untuk menyederhanakan hasilnya secara simbolik dengan menggunakan perintah ev dan opsi simpsum=true.
sum(k, k, n, inf): Ini adalah ekspresi yang digunakan untuk menghitung jumlah deret tak hingga

$$\sum_{k=n}^{\infty} k$$

dengan x sebagai variabel indeks, dimulai dari x=n hingga tak terhingga(inf). Ini adalah notasi umum untuk menggambarkan deret tak hingga yang dimulai dari n hingga tak terhingga.
=: Ini adalah operator yang digunakan untuk menetapkan hasil perhitungan di sebelah kiri kepada ekspresi di sebelah kanan. Dalam hal ini, pernyataan ini mengatakan bahwa hasil dari perhitungan deret sebelah kiri harus sama dengan hasil dari perhitungan deret sebelah kanan.
ev(...): Ini adalah perintah dalam bahasa yang digunakan untuk mengevaluasi (menghitung) ekspresi matematika.
simpsum=true: Ini adalah opsi yang digunakan bersamaan dengan perintah ev untuk memberitahu bahwa Anda ingin mencoba menyederhanakan hasil perhitungan deret secara simbolik jika memungkinkan. Dengan kata lain, opsi ini menginstruksikan untuk melakukan penyederhanaan pada hasil deret.
Macam-Macam Deret Tak Hingga
1. Deret Geometri
Deret dengan suku-suku yang memiliki rasio tetap antara setiap dua suku berturut-turut.
Rumus Deret Geometri

$$S_\infty = \frac{a}{1-r}$$

```
>$'sum(1/2^k, k, 1, inf)= ev(sum(1/2^k, k, 1, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{k=1}^{\infty} \frac{1}{2^k} = 1$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum_{k=n}^{\infty} \frac{1}{2^k}$$

Deret tersebut adalah deret geometri dengan suku awal a=1
dan rasio r=1/2. Deret geometri ini memiliki susku-suku

$$\frac{1}{2^k}$$

yang dimulai dari k=1 hingga tak terhingga. Hasil dari perhitungan tersebut adalah 1, karena itu adalah jumlah dari deret geometri yang mendekati setengah dari 1 ketika tak terhingga.

```
>$'sum(1/5^x, x, 2, inf)= ev(sum(1/5^x, x, 2, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=2}^{\infty} \frac{1}{5^x} = \frac{1}{20}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum \frac{1}{5^x}$$

Deret tersebut adalah deret geometri dengan suku awal

$$a = \frac{1}{5^2}$$

(karena dimulai dari x=2)
dan rasio r=1/5. Deret geometri ini memiliki susku-suku

$$\frac{1}{5^x}$$

yang dimulai dari x=2 hingga tak terhingga.Hasil dari perhitungan tersebut adalah 1/20, karena itu adalah jumlah dari deret geometri yang mendekati setengah dari 1/20 ketika tak terhingga.

```
>$'sum(1/2^x, x, 3, inf)= ev(sum(1/2^x, x, 3, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=3}^{\infty} \frac{1}{2^x} = \frac{1}{4}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum \frac{1}{2^x}$$

Deret tersebut adalah deret geometri dengan suku awal

$$a = \frac{1}{2^3}$$

(karena dimulai dari x=3)
dan rasio r=1/2. Deret geometri ini memiliki susku-suku

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

yang dimulai dari x=1 hingga tak terhingga. Hasil dari perhitungan tersebut adalah 1/4, karena itu adalah jumlah dari deret geometri yang mendekati setengah dari 1/4 ketika tak terhingga.
2. Deret Harmonik
Deret dengan suku-suku yang memiliki bentuk

$$\frac{1}{n}$$

di mana n adalah bilangan bulat positif
3. Deret Harmonik Kuadrat Terbalik
Deret dengan suku-suku yang memiliki bentuk

$$\frac{1}{n^2}$$

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum \frac{1}{x^2}$$

Deret tersebut adalah deret harmonik kuadrat terbalik dengan suku awal a=1 dan rasio yang tidak konstan. Deret harmonik kuadrat terbalik ini memiliki susku-suku

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

yang dimulai dari x=1 hingga tak terhingga. Hasil dari perhitungan tersebut adalah

$$\frac{\pi^2}{6}$$

```
>$'sum(1/x^2, x, 2, inf)= ev(sum(1/x^2, x, 2, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=2}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6} - 1$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum \frac{1}{x^2}$$

Deret tersebut adalah deret harmonik kuadrat terbalik dengan suku awal a=1/4 dan rasio yang tidak konstan. Deret harmonik kuadrat terbalik ini memiliki susku-suku

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

yang dimulai dari x=2 hingga tak terhingga. Hasil dari perhitungan tersebut adalah

$$\frac{\pi^2}{6} - 1$$

```
>$'sum(1/x^2, x, 3, inf)= ev(sum(1/x^2, x, 3, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=3}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6} - \frac{5}{4}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum \frac{1}{x^2}$$

Deret tersebut adalah deret harmonik kuadrat terbalik dengan suku awal a=1/9 dan rasio yang tidak konstan. Deret harmonik kuadrat terbalik ini memiliki susku-suku

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

yang dimulai dari x=3 hingga tak terhingga. Hasil dari perhitungan tersebut adalah

$$\frac{\pi^2}{6} - \frac{5}{4}$$

4. Deret Eksponensial
Deret yang yang memiliki suku-suku yang didefinisikan oleh fungsi eksponensial

$$e^x$$

```
>$'sum(exp(-x), x, 1, inf) = ev(sum(exp(-x), x, 1, inf), simpsum=true) // ev: menghitung n
```

$$\sum_{x=1}^{\infty} e^{-x} = \frac{e^{-1}}{1 - e^{-1}}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum e^{-x}$$

Deret tersebut adalah deret eksponensial dengan suku awal

$$a = e^{-1}$$

dan rasio

$$a = e^{-1}$$

yang dimulai dari x=1 hingga tak terhingga. Hasil dari perhitungan tersebut adalah

$$\frac{e^{-1}}{1-e^{-1}}$$

```
>$'sum(exp(-2*x), x, 1, inf) = ev(sum(exp(-2*x), x, 1, inf), simpsum=true) // ev: menghitu
```

$$\sum_{x=1}^{\infty} e^{-2\,x}=\frac{e^{-2}}{1-e^{-2}}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum e^{-2x}$$

Deret tersebut adalah deret eksponensial dengan suku awal

$$a=e^{-2}$$

dan rasio

$$a=e^{-2}$$

yang dimulai dari x=1 hingga tak terhingga. Hasil dari perhitungan tersebut adalah

$$\frac{e^{-2}}{1-e^{-2}}$$

```
>$'sum(exp(-x), x, 2, inf) = ev(sum(exp(-x), x, 2, inf), simpsum=true) // ev: menghitung n
```

$$\sum_{x=2}^{\infty} e^{-x}=\frac{e^{-2}}{1-e^{-1}}$$

Perintah tersebut menghitung jumlah deret tak hingga

$$\sum e^{-x}$$

Deret tersebut adalah deret eksponensial dengan suku awal

$$a=e^{-2}$$

dan rasio

$$a=e^{-1}$$

yang dimulai dari x=2 hingga tak terhingga. Hasil dari perhitungan tersebut adalah

$$\frac{e^{-2}}{1-e^{-1}}$$

5. Deret Taylor
Deret taylor adalah deret yang dalam bentuk deret tak hingga yang terdiri dari suku-suku berurutan. Deret ini digunakan untuk mendekati fungsi matematis dengan menggunakan suku-suku deret. Deret Taylor umumnya digunakan untuk melakukan ekspansi fungsi di sekitar suatu titik tertentu dalam bentuk deret tak hingga. Deret Taylor digunakan untuk menentukan nilai hampiran dari fungsi dalam bentuk polinom.
Deret Taylor dari suatu fungsi f(x) di sekitar titik a memiliki bentuk umum seperti berikut:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + ...$$

Setiap suku dalam deret Taylor ini bergantung pada turunan fungsi f(x) di titik a. Deret Taylor dapat digunakan untuk mendekati nilai fungsi f(x) di sekitar titik a. Semakin banyak suku yang digunakan dalam deret Taylor, semakin akurat pendekatannya.
Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$'f(sin(x)) =taylor(sin(x), x, 0, 6) // deret Taylor f(sin(x)) di sekitar x=0, sampai ord
```

$$f(\sin x) = \frac{x^5}{120} - \frac{x^3}{6} + x$$

Perintah ini adalah contoh penggunaan deret Taylor untuk mendekati fungsi f(x)di sekitar x=0 hingga orde 6. Ini akan menghasilkan deret Taylor dari f(x)yang mencakup suku-suku hingga orde 6. Hasil dari perintah ini adalah deret Taylor dari f(x). Deret ini dapat digunakan untuk mendekati nilai fungsi di sekitar x=0 dengan tingkat keakuratan yang diinginkan.

```
>$'f(cos(x)) =taylor(cos(x), x, pi, 4) // deret Taylor f(sin(x)) di sekitar x=pi, sampai o
```

$$f(\cos x) = -\frac{(x-\pi)^4}{24} + \frac{(x-\pi)^2}{2} - 1$$

Perintah ini adalah contoh penggunaan deret Taylor untuk mendekati fungsi f(cos(x))di sekitar

$$x = \pi$$

hingga orde 6. Ini akan menghasilkan deret Taylor dari f(x)yang mencakup suku-suku hingga orde 6. Hasil dari perintah ini adalah deret Taylor dari f(x). Deret ini dapat digunakan untuk mendekati nilai fungsi di sekitar

$$x = \pi$$

dengan tingkat keakuratan yang diinginkan.

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai orde ke-10
```

$$e^x = \frac{x^{10}}{3628800} + \frac{x^9}{362880} + \frac{x^8}{40320} + \frac{x^7}{5040} + \frac{x^6}{720} + \frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

Perintah ini adalah contoh penggunaan deret Taylor untuk mendekati fungsi eksponensial

$$e^x$$

di sekitar x=0 hingga orde 10. Ini akan menghasilkan deret Taylor dari

$$e^x$$

yang mencakup suku-suku hingga orde 10. Hasil dari perintah ini adalah deret Taylor dari

$$e^x$$

Deret ini dapat digunakan untuk mendekati nilai fungsi di sekitar x=0 dengan tingkat keakuratan yang diinginkan.

```
>$'log(x) =taylor(log(x), x, 1, 10) // deret Taylor log(x) di sekitar x=1, sampai orde ke-
```

$$\log x = x - \frac{(x-1)^{10}}{10} + \frac{(x-1)^9}{9} - \frac{(x-1)^8}{8} + \frac{(x-1)^7}{7} - \frac{(x-1)^6}{6} + \frac{(x-1)^5}{5} - \frac{(x-1)^4}{4} + \frac{(x-1)^3}{3} - \frac{(x-1)^2}{2} - 1$$

Perintah ini adalah contoh penggunaan deret Taylor untuk mendekati fungsi logaritma log(x) di sekitar x=0 hingga orde 10. Ini akan menghasilkan deret Taylor dari log(x) yang mencakup suku-suku hingga orde 10. Hasil dari perintah ini adalah deret Taylor dari log(x). Deret ini dapat digunakan untuk mendekati nilai fungsi di sekitar x=0 dengan tingkat keakuratan yang diinginkan.

```
>$'f(sqrt(x)) =taylor(sqrt(x), x, 4, 5) // deret Taylor f(sqrt(x)) di sekitar x=4, sampai
```

$$f\left(\sqrt{x}\right) = \frac{x-4}{4} + \frac{7\,(x-4)^5}{131072} - \frac{5\,(x-4)^4}{16384} + \frac{(x-4)^3}{512} - \frac{(x-4)^2}{64} + 2$$

Kekonvergenan

Kekonvergenan adalah salah satu konsep penting dalam matematika yang digunakan untuk menggambarkan perilaku deret atau barisan. Kekonvergenan mengacu pada sifat deret atau barisan yang mendekati nilai tertentu saat jumlah suku atau indeks barisan mendekati tak hingga.

Dengan kata lain, konvergenan mengacu pada apakah deret atau barisan tersebut memiliki "limit" atau "jumlah" saat suku-sukunya diperbanyak hingga tak terbatas.

Konsep utama dalam kekonvergenan

1. Kekonvergenan Deret

Deret adalah penjumlahan tak hingga suku-suku barisan. Kekonvergenan deret mengacu pada sifat deret yang konvergen, yang berarti bahwa deret tersebut memiliki jumlah yang terbatas saat jumlah suku deret meningkat tak hingga. Jika deret konvergen, maka jumlahnya disebut "konvergen deret" atau "limit deret." Deret yang konvergen adalah deret yang jumlahnya ada dan terhingga.

Deret dikatakan konvergen jika, seiring kita menambahkan lebih banyak suku, jumlah parsial deret mendekati (dalam arti mendekati dalam batas tertentu) nilai L. Jika kondisi ini terpenuhi, maka deret tersebut dikatakan konvergen dan memiliki hasil yang konvergen, yaitu L.

2. Kekonvergenan Barisan

Barisan adalah urutan suku-suku yang tidak dijumlahkan. Kekonvergenan barisan mengacu pada sifat barisan yang mendekati nilai tertentu saat indeks barisan meningkat tak hingga. Dalam hal ini, kita tidak mencari jumlah, tetapi kita ingin menentukan apakah barisan memiliki "limit" saat indeks mendekati tak hingga. Jika barisan memiliki limit yang terhingga, maka barisan disebut "konvergen barisan"

Dalam hal ini, kita memeriksa apakah suku-suku barisan an mendekati (dalam arti mendekati dalam batas tertentu) nilai L saat n meningkat tak terbatas. Jika kondisi ini terpenuhi, maka barisan tersebut dikatakan konvergen dan memiliki hasil yang konvergen, yaitu L.

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1
```

```
0.739085133216
```

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = cos(x)$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

```
~0.739085133211,0.7390851332133~
```

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan

$$x = cos(x)$$

```
>h=expand(hasil,100), cos(h) << h
```

```
~0.73908513309,0.73908513333~
1
```

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi x(n+1)=f(x(n)) akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

```
1.41421356237
1.41421356237
```

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

```
>iterate("f",2,5)
```

```
[2,  1.5,  1.41667,  1.41422,  1.41421,  1.41421]
```

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",~1,2~,5)
```

```
[ ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~ ]
```

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

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

Satu-satunya solusi adalah

$$\sqrt{2}$$

```
>iterate("s",~1,2~)
```

```
~1.41421356236,1.41421356239~
```

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1},b_{n+1})=\left(\frac{a_n+b_n}{2},\sqrt{a_nb_n}\right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2;sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g",[1;5])
```

```
      2.60401
      2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g",[1;5],4)
```

```
            1             3       2.61803       2.60403       2.60401
            5       2.23607       2.59002       2.60399       2.60401
```

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g",[~1~;~5~],4)
```

```
Interval 2 x 5 matrix

~0.99999999999999778,1.00000000000000022~      ...
~4.99999999999999911,5.00000000000000089~      ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)",2,10)
```

```
[2,  1.41421,  1.18921,  1.09051,  1.04427,  1.0219,  1.01089,
1.00543,  1.00271,  1.00135,  1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)
```

```
[1.04888,  1.00028,  1,  1]
```

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

```
1.41421356237
```

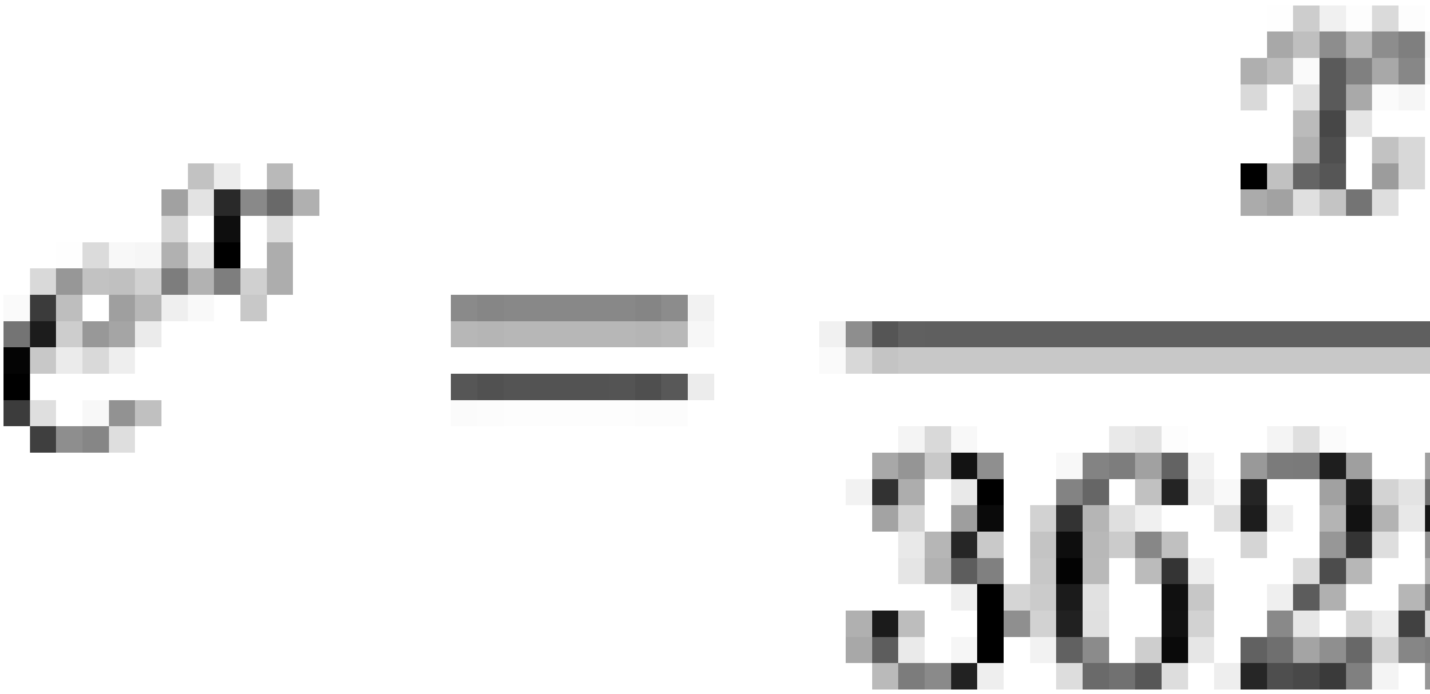
Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

```
[1,  2,  6,  24,  120,  720,  5040,  40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
>plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
```



```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
>x=[1;1]; repeat xnew=A.x-b; until all(xnew~=x); x=xnew; end; ...
>x,
```

```
        -7.09677
        -7.74194
```

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.
Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
```

```
  endfunction
```

Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil
```

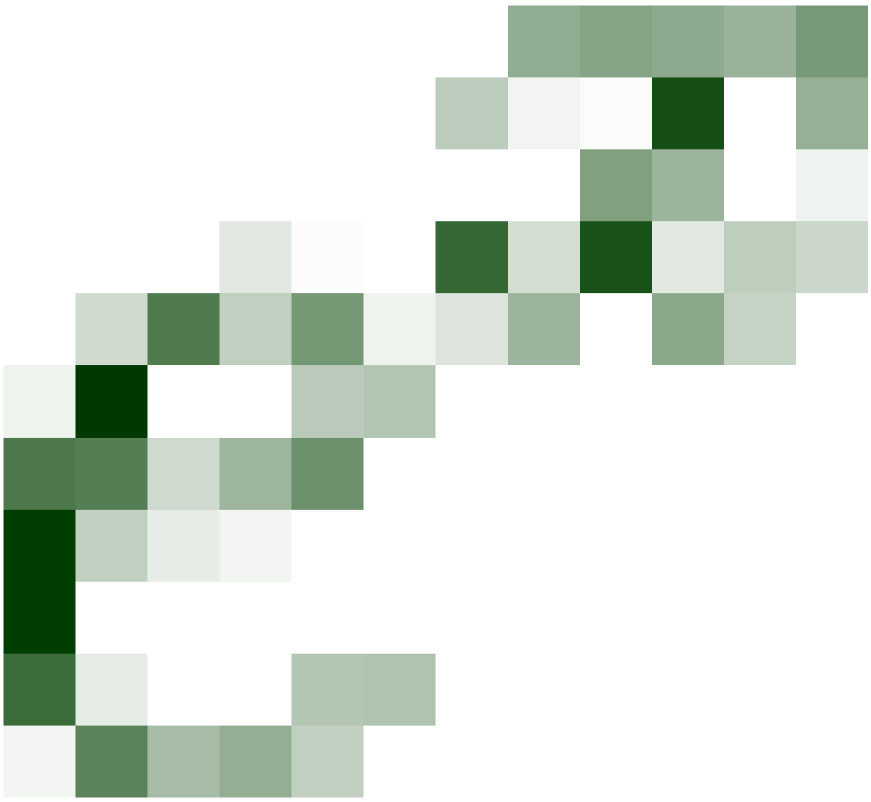
```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

$$\left[1+x , 1+x+\frac{x^2}{2} , 1+x+\frac{x^2}{2}+\frac{x^3}{6}\right]$$

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor eskpresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi ters
```

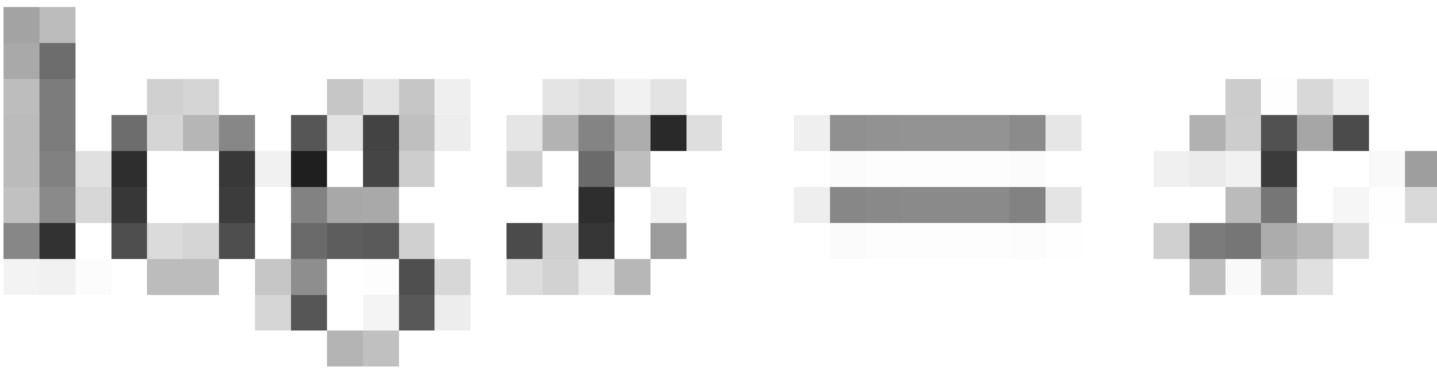


Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

$$1+x+\frac{x^2}{2}+\frac{x^3}{6}$$

```
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
```



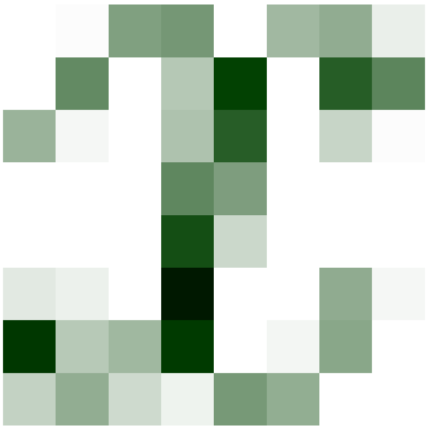
```
>$sum(sin(k*x)/k,k,1,5)
```

$$\sin x+\frac{\sin\left(2\,x\right)}{2}+\frac{\sin\left(3\,x\right)}{3}+\frac{\sin\left(4\,x\right)}{4}+\frac{\sin\left(5\,x\right)}{5}$$

Berikut adalah cara menggambar kurva

$$y=\sin(x)+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots.$$

```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```



Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \le x \le 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```

BAB 5

KB PEKAN 8: MENGGUNAKAN EMT UNTUK GEOMETRI

[a4paper,10pt]article eumat

Akhmad Rizki Khamid
22305141008
Matematika E

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 geometeri 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 <ABC
angleBisector(A, B, C): garis bagi sudut <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 lingkran 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 dengan alpha sudut yang menghadap sisi a.

```
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga
```

dengan panjang sisi a, b, c.

```
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk
```

suatu segitiga

```
doublespread(sa): Spread sudut rangkap Spread 2*phi, dengan
```

sa=sin(phi)^2 spread a.

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 atur tiga poin dan plot.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
```

Lalu 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 untuk garis adalah [a, b, c], yang merepresentasikan garis
dengan persamaan ax + by = c

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1,  2,  2]
```

Hitung garis tegak lurus melalui A pada BC.

```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan perpotongannya 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...()
```

images/EMTGeometri_22305141008_Akhmad R.K-001.png

Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2}AD.BC.$$

```
>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 langusng dengan fungsi
```

```
1.5
```

Cara lain menghitung luas segitigas ABC:

```
>distance(A,D)*distance(B,C)/2
```

```
1.5
```

sudut di C.

```
>degprint(computeAngle(B,C,A))
```

```
36°52'11.63''
```

Sekarang lingkaran sirkit 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"):
```

images/EMTGeometri_22305141008_Akhmad R.K-002.png

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
```

images/EMTGeometri_22305141008_Akhmad R.K-003.png

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.
3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
4. Gambar jari-jari lingkaran dalam.

Jawab :
1.) Titik singgung BC dengan lingkaran dalam

```
>s=lineThrough(B,C)
```

```
[-1,  2,  2]
```

```
>m=circleWithCenter(P,r)
```

```
[0.86038,  0.86038,  0.509654]
```

```
>S=lineCircleIntersections(s,m)
```

```
[0.632456,  1.31623]
```

Titik singgung garis AC dengan lingkaran dalam

```
>p=lineThrough(A,C)
```

```
[-2,  1,  -2]
```

```
>Q=lineCircleIntersections(p,m)
```

```
[1.31623,  0.632456]
```

```
>q=lineThrough(A,B)
```

```
[-1,  -1,  -1]
```

```
>L=lineCircleIntersections(q,m)
```

```
[0.5,  0.5]
```

jadi titik singgung lingkaran dalam dengan sisi-sisi segitiga adalah
(0.632456, 1.31623) , (1.31623,0.632456), dan (0.5,0.5)
2.)

```
>plotSegment(S,Q,"a");
>plotSegment(S,L,"b");
>plotSegment(L,Q,"c"):
```

images/EMTGeometri_22305141008_Akhmad R.K-004.png

3.)

```
>P, r
```

```
[0.86038,  0.86038]
0.509653732104
```

```
>k=angleBisector(A,B,C)
```

```
[-0.264911,  -1.63246,  -1.63246]
```

```
>color(3); plotLine(k):
```

images/EMTGeometri_22305141008_Akhmad R.K-005.png

```
>plotSegment(P,L,"r"):
```

images/EMTGeometri_22305141008_Akhmad R.K-006.png

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri tepat dan simbolis menggunakan Maxima.
Geometri file.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, sekarang kita dapat menggunakan perhitungan simbolik.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan penghitungan simbolik.

```
>c &= lineThrough(B,C) // c=BC
```

```
[- 1, 2, 2]
```

Kita bisa mendapatkan persamaan untuk sebuah garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$\left[y=\frac{x}{2}+1\right]$$

images/EMTGeometri_22305141008_Akhmad R.K-008-larg

```
>$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
```

```
 2   6
[-, -]
 5   5
```

```
>$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
```

$$1.178511301977579$$

images/EMTGeometri_22305141008_Akhmad R.K-014-lar

```
>$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 bisa 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 mengembalikan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2,
```

```
[4.64575,  -1.64575]
[-0.645751,  3.64575]
```

```
>plotPoint(P1); plotPoint(P2):
```

images/EMTGeometri_22305141008_Akhmad R.K-018.png

Hal yang sama 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);
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-020.png

Garis Sumbu

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 melallui 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):
```

images/EMTGeometri_22305141008_Akhmad R.K-021.png

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. Tapi kita bisa menyederhanakan, jika kita menyelesaikan y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y = \frac{-(2\,b_1 - 2\,a_1)\,x + {b_2}^2 + {b_1}^2 - {a_2}^2 - {a_1}^2}{2\,b_2 - 2\,a_2}\right]$$

Ini memang sama dengan tengah tegak lurus, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y=\frac{\left(b_2-a_2\right)\,x-a_1\,b_2+a_2\,b_1}{b_1-a_1}\right]$$

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 larutan 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 formula Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H\left(a , b , \left[1 , 0 , 4 \right] \right)=\frac{\left| a\right| \,\left| ysol\right| }{2}$$

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 or function ysol not found.
Try "trace errors" to inspect local variables after errors.
H:
    useglobal; return abs(a)*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
        ^
```

Dan jelas juga, bahwa ini adalah segitiga dengan luas maksimal dan kedua sisinya 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga sengan panjang sisi 3, 4, x (
```

```
Variable or function ysol not found.
Error in expression: 3*abs(ysol)/2
 %ploteval:
    y0=f$(x[1],args());
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
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 elips.

```
>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 manfaatkan 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
```

$$0$$

images/EMTGeometri_22305141008_Akhmad R.K-027-lar

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```

images/EMTGeometri_22305141008_Akhmad R.K-028.png

Kita dapat memeriksa persamaan umum elips ini, yaitu.

$$\frac{(x-x_m)^2}{u^2}+\frac{(y-y_m)}{v^2}=1,$$

di mana (xm, ym) 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])
```

$$\frac{a^2}{d^2}$$

Kita melihat bahwa tinggi dan luas segitiga adalah maksimal untuk x = 0. Jadi luas segitiga dengan a + b + c = d adalah maksimal, jika sama sisi. Kami ingin mendapatkan 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
```

$$\left[\frac{a\,ysol^2}{2}=0,0=0\right]$$

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusi a = b = c = d / 3.

```
>$solve(eqns,[a,b])
```

$$[[a=0,b=\%r_1]]$$

Ada juga metode Lagrange, memaksimalkan H (a, b, c) ^ 2 terhadap a + b + d = 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:
  ... la,    diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...
                                                       ^
```

Kita bisa membuat plot situasinya

Pertama, 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
```

$$[a,0]$$

images/EMTGeometri_22305141008_Akhmad R.K-033-lar

Kemudian atur rentang plot, dan plot poinnya.

```
>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.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
... otPoint(mxmeval("C"),"C"); plotPoint(mxmeval("A"),"A"): ...
                                                      ^
```

Plot segmennya.

```
>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 tengah tegak lurus di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan bagian tengah dari keliling.

```
>U &= lineIntersection(h,g);
```

Kami mendapatkan rumus untuk jari-jari lingkaran sunat.

```
>&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()); ...
>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 mendapatkan rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radius. Kami dapat memeriksa, apakah ini benar dengan Maxima. Maxima akan memfaktorkannya hanya jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))^2  | factor
```

$$\left[\frac{a_2^2+a_1^2}{a_2^2}, 0, \frac{16\left(a_2^2+a_1^2\right)}{a_2^2}\right]$$

Contoh 4: Garis Euler dan Parabola

Garis euler adalah garis yang ditentukan dari segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk pusat ortosentrum, sirkumenter, pusat massa, titik Exeter, dan pusat lingkaran sembilan titik segitiga.
Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam segitiga.
Pertama, kami menentukan sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolik.

```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan poin ke dalamnya. 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,""):
```

images/EMTGeometri_22305141008_Akhmad R.K-036.png

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 dari sisi c.

```
>c &= lineThrough(A,B)
```

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

Dan juga dapatkan rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

$$3\,y-x=-2$$

Untuk bentuk Hesse, kita perlu menentukan 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]])
```

$$\frac{7}{\sqrt{10}}$$

images/EMTGeometri_22305141008_Akhmad R.K-040-larg

Sekarang kami menghitung sirkit 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
```

$$\left[\frac{3}{14},\frac{5}{14}\right]$$

Plot lingkaran dan pusatnya. Cu dan U adalah simbolik. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```

images/EMTGeometri_22305141008_Akhmad R.K-043.png

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 tersebut.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19\,y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kita.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

images/EMTGeometri_22305141008_Akhmad R.K-046.png

Pusat gravitasi harus berada di garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M"): // titik berat
```

images/EMTGeometri_22305141008_Akhmad R.K-048.png

Teorinya mengatakan bahwa MH = 2 * MO. Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
>$distance(M,H)/distance(M,O)|radcan
```

$$2$$

Fungsinya termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\,\sqrt{13}}\right)$$

```
60°15'18.43''
```

Persamaan untuk pusat lingkaran tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

$$\left[\frac{\left(2^{\frac{3}{2}}+1\right)\,\sqrt{5}\,\sqrt{13}-15\,\sqrt{2}+3}{14} , \frac{\left(\sqrt{2}-3\right)\,\sqrt{5}\,\sqrt{13}+5\,2^{\frac{3}{2}}+5}{14}\right]$$

Mari kita hitung juga ekspresi jari-jari lingkaran yang tertulis.

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{\left(-41\,\sqrt{2}-31\right)\,\sqrt{5}\,\sqrt{13}+115\,\sqrt{2}+614}}{7\,\sqrt{2}}$$

```
>LD &=  circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
>color(5); plotCircle(LD()):
```

images/EMTGeometri_22305141008_Akhmad R.K-053.png

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
```

$$\frac{3\,y-x+2}{\sqrt{10}}-\sqrt{\left(2-y\right)^2+\left(1-x\right)^2}=0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```

images/EMTGeometri_22305141008_Akhmad R.K-055.png

This should be some function, but the default solver of Maxima can find the solution only, if we square the equation. Consequently, we get a fake solution.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
                      y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

The first solution is
maxima: akar[1]
Adding the first solution to the plot show, that it is indeed the path we are looking for. The theory tells us that it is a rotated parabola.

```
>plot2d(&rhs(akar[1]),add=1):
```

images/EMTGeometri_22305141008_Akhmad R.K-056.png

```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g\left(x\right)=-3\,x-\sqrt{70}\,\sqrt{9-2\,x}+26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$2.135605779339061$$

images/EMTGeometri_22305141008_Akhmad R.K-059-lar

```
>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(%) // jatak T ke AB
```

$$2.135605779339061$$

images/EMTGeometri_22305141008_Akhmad R.K-062-lar

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 oleh ceramah N.J.Wildberger. Dalam bukunya "Proporsi Agung", Wildberger mengusulkan untuk menggantikan pengertian klasik tentang jarak dan sudut dengan kuadransi 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 dapat dilakukan dengan kertas dan pensil saja. 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 mengevaluasi ke pendekatan numerik saja.

```
>load geometry;
```

Untuk pendahuluan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk memplot 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);
```

images/EMTGeometri_22305141008_Akhmad R.K-063.png

Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana wa adalah sudut di A. Cara biasa untuk menghitung sudut ini, adalah dengan melakukan invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak secara perkiraan.

```
>wa := arcsin(3/5); degprint(wa)
```

```
36°52'11.63''
```

Trigonometri rasional mencoba menghindari hal ini.
Pengertian pertama dari trigonometri rasional adalah kuadran, yang menggantikan jarak. Faktanya, itu hanyalah kuadrat jarak. Berikut ini, a, b, dan c menunjukkan kuadran sisi-sisinya.
Teorema Pythogoras menjadi a + b = c lalu.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

```
25 = 25
```

Gagasan kedua dari trigonometri rasional adalah penyebarannya. Spread mengukur bukaan antar baris. Ini adalah 0, jika garis sejajar, dan 1, jika garis persegi panjang. Ini adalah kuadrat dari 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 segitiga persegi panjang mana pun dengan satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti yang sudut dapat ditambahkan dengan mudah.
Tentu saja, kita dapat mengubah nilai perkiraan sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

```
9/25
```

Hukum kosinus dari trgonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c + b - a)^2 = 4bc\,(1 - s_a)$$

Di sini a, b, dan c adalah kuadran dari sisi-sisi segitiga, dan sa adalah sebaran di sudut A. Sisi a, seperti biasa, berlawanan dengan sudut A.
Hukum ini diimplementasikan dalam file geometry.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[\left(bb-aa+\frac{7}{6}\right)^2, \left(bb-aa+\frac{7}{6}\right)^2, \left(bb-aa+\frac{5}{3\sqrt{2}}\right)^2\right] = \left[\frac{14\,bb\,(1-saa)}{3}, \frac{14\,bb\,(1-saa)}{3}, \frac{5\,2^{\frac{3}{2}}\,bb\,(1-saa)}{3}\right]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari sebaran di A. Untuk melakukan ini, kita menghasilkan crosslaw untuk kuadran a, b, dan c, dan menyelesaikannya untuk sebaran yang tidak diketahui sa.
Anda dapat melakukan ini dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah mendapatkannya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$\left[x = \frac{9}{25}\right]$$

images/EMTGeometri_22305141008_Akhmad R.K-068-larg

Kami sudah tahu ini. Definisi penyebaran adalah kasus khusus dari hukum lintas hukum.
Kita juga bisa menyelesaikan ini untuk umum a, b, c. Hasilnya adalah rumus yang menghitung sebaran sudut segitiga berdasarkan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[\left[\frac{168\,bb\,x+36\,bb^2+(-72\,aa-84)\,bb+36\,aa^2-84\,aa+49}{36}, \frac{168\,bb\,x+36\,bb^2+(-72\,aa-84)\,bb+36\,aa^2-84\,aa+49}{36},\right.\right.$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah ditentukan dalam file geometry.e Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita bisa menggunakannya untuk menghitung sudut segitiga bersisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

```
67°47'32.44''
```

Contoh lain

Sekarang, mari kita coba contoh yang lebih canggih.
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;
```

images/EMTGeometri_22305141008_Akhmad R.K-072.png

Menggunakan Pythogoras, 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 memiliki fungsi kuadrans antara dua titik.
Dalam contoh berikut, karena c + b bukan a, segitiga tidak persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$17$$

images/EMTGeometri_22305141008_Akhmad R.K-075-lar

images/EMTGeometri_22305141008_Akhmad R.K-076-lar

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan perkalian titik dari dua vektor. Hasilnya adalah beberapa pendekatan floating point.

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\,\sqrt{17}}\right)$$

```
32.4711922908
```

Menggunakan pensil dan kertas, kita bisa melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan untuk x.

```
>$crosslaw(a,b,c,x), $solve(%,x),
```

$$\left[x = \frac{49}{50}\right]$$

images/EMTGeometri_22305141008_Akhmad R.K-079-larg

Artinya, fungsi penyebaran yang didefinisikan dalam "geometry.e" tidak.

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Itu menyelesaikan istilah sin (arccos (...)) menjadi hasil pecahan. Kebanyakan siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita mendapatkan sebaran di B, kita bisa 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 diproduksi dengan program geometri C.a.R., yang dapat menggambar kuadran dan menyebar.
gambar : (20) Rational_Geometry_CaR.png
Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, bahwa kita berurusan dengan kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan yang biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

The Heron Formula

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama-tama kita menghitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kami menghitung luas area yang dikuadratkan ("kuadrea"?), Memfaktorkannya dengan Maxima, dan kami 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)
```

$$\frac{(-c+b+a)\ (c-b+a)\ (c+b-a)\ (c+b+a)}{16}$$

images/EMTGeometri_22305141008_Akhmad R.K-087-larg

The Triple Spread Rule

Kerugian dari spread adalah bahwa mereka tidak lagi hanya menambahkan sudut serupa.
Namun, tiga sebaran segitiga memenuhi aturan "penyebaran rangkap tiga" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc+sb+sa)^2=2\left(sc^2+sb^2+sa^2\right)+4\,sa\,sb\,sc$$

Aturan ini berlaku untuk tiga sudut yang bertambah menjadi 180 ?.

$$\alpha+\beta+\gamma=\pi$$

Sejak penyebaran

$$\alpha,\pi-\alpha$$

sama, aturan penyebaran tiga kali lipat juga benar, jika

$$\alpha+\beta=\gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran tiga kali lipat juga berlaku, jika

$$\alpha+\beta+\gamma=0$$

Misalnya, kita dapat menghitung sebaran sudut 60 ?. Ini 3/4. Persamaan memiliki solusi kedua, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[x=\frac{3}{4},x=0\right]$$

Sebaran 90 ? jelaslah 1. Jika dua sudut dijumlahkan menjadi 90 ?, penyebarannya menyelesaikan persamaan penyebaran rangkap tiga dengan a, b, 1. Dengan perhitungan berikut kita mendapatkan a + b = 1.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$[x=1-y]$$

images/EMTGeometri_22305141008_Akhmad R.K-091-lar

Karena penyebaran 180 ? -t sama dengan penyebaran t, rumus penyebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau perbedaan dari dua sudut lainnya.
Jadi kita bisa menemukan sebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami menjadikan ini sebuah fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

$$\left[x=4\,a-4\,a^2,x=0\right]$$

```
- 4 (a - 1) a
```

Angle Bisectors

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;
```

images/EMTGeometri_22305141008_Akhmad R.K-093.png

Mari kita hitung panjang bisektor sudut pada A. Tapi kita ingin menyelesaikannya untuk umum a, b, c.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita menghitung sebaran sudut terbagi di A, menggunakan rumus sebaran rangkap tiga. Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kami harus memilih yang benar. Solusi lainnya mengacu pada sudut terbagi 180 ? -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

images/EMTGeometri_22305141008_Akhmad R.K-095-large

images/EMTGeometri_22305141008_Akhmad R.K-096-larg

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kami dapat mencetak sudut di Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

```
18°26'5.82''
```

Titik P adalah perpotongan dari garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
[0,  1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```

images/EMTGeometri_22305141008_Akhmad R.K-098.png

Mari kita periksa sudut dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
0.321750554397
```

Sekarang kita menghitung panjang bisektor AP.
Kami menggunakan teorema sinus di segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

memegang di segitiga apa pun. Persegi itu, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

dimana a, b, c menunjukkan qudrance.
Karena BPA sebaran adalah 1-sa2, kita dapatkan darinya bisa / 1 = b / (1-sa2) dan dapat menghitung bisa (kuadran garis-garis).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2\,b\,\left(b+a\right)}{\sqrt{b}\,\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356
4.21637021356
```

Kami 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
```

The Chord Angle

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,"0"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-101.png

Kita bisa menggunakan Maxima untuk menyelesaikan rumus sebaran rangkap tiga untuk sudut di pusat O untuk r. Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari keliling dalam hal kuadrat sisi.
Kali ini, Maxima menghasilkan beberapa angka nol yang 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
```

$$-\frac{a\,b\,c}{c^2-2\,b\,c+a\,\left(-2\,c-2\,b\right)+b^2+a^2}$$

Kita bisa menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk poin A, B, C kita.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

```
>periradius(a,b,c)
```

```
1
```

Faktanya, penyebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut akord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Sebenarnya sebarannya adalah b / (4r), dan kita melihat bahwa sudut akor b adalah setengah dari sudut tengah.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

$$0$$

Contoh 6: Jarak Minimal pada Bidang

Ucapan 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"):
```

images/EMTGeometri_22305141008_Akhmad R.K-105.png

dan grafiknya juga agak sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

images/EMTGeometri_22305141008_Akhmad R.K-106.png

Tentu saja minimal 0 dicapai di A.

Two points

Sekarang kita melihat fungsi MA + MB dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang terkenal" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk minimum AB 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):
```

images/EMTGeometri_22305141008_Akhmad R.K-107.png

Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

images/EMTGeometri_22305141008_Akhmad R.K-108.png

Batasan ke baris (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

images/EMTGeometri_22305141008_Akhmad R.K-109.png

Three points

Sekarang hal-hal menjadi kurang sederhana: Sedikit kurang diketahui bahwa MA + MB + MC mencapai minimumnya pada satu titik bidang tetapi untuk menentukannya kurang sederhana:
1) Jika salah satu sudut segitiga ABC lebih dari 120 ? (katakanlah dalam A), maka minimum tercapai pada titik ini (katakanlah 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);
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-110.png

```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-111.png

2) Tetapi jika semua sudut segitiga ABC kurang dari 120 ?, minimum berada pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi ABC dengan sudut yang sama (lalu masing-masing 120) :

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```

images/EMTGeometri_22305141008_Akhmad R.K-112.png

```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-113.png

Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; sebagai contoh, saya tahu soft tertulis di Java yang memiliki instruksi "garis kontur" ...
Semua ini di atas telah ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; dia menulis surat kepada para penggila lainnya seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di bagian pajak penghasilan. Jadi titik unik F sehingga FA + FB + FC minimal disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torriccelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Pokoknya tradisinya adalah mencatat poin ini ...

Four points

Langkah selanjutnya adalah menambahkan titik D ke-4 dan mencoba meminimalkan MA + MB + MC + MD; katakanlah 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 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):
```

images/EMTGeometri_22305141008_Akhmad R.K-114.png

```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-115.png

Masih ada minimum dan tidak ada yang dicapai pada simpul A, B, C atau D:

```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
[0.142858,  0.142857]
```

Tampaknya dalam kasus ini, koordinat titik optimal rasional atau mendekati rasional ...
Sekarang ABCD adalah bujur sangkar, kami berharap bahwa titik optimal adalah pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```

images/EMTGeometri_22305141008_Akhmad R.K-116.png

```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-117.png

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda memiliki Povray diinstal, dan pvengine.exe di jalur program.
Pertama kami menghitung 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 geometry.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]
```

Lalu 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(),"")
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```

images/EMTGeometri_22305141008_Akhmad R.K-118.png

Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[0, u]
```

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2\,u}{a^2+1}-u\right)^2+\frac{a^2\,u^2}{\left(a^2+1\right)^2}}$$

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u+2}{5}-u\right)^2+\frac{\left(2\,u-1\right)^2}{25}}$$

Dan temukan pusat kedua lingkaran, di mana jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u=\frac{-\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1} , u=\frac{\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1}\right]$$

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]),"");
>insimg;
```

images/EMTGeometri_22305141008_Akhmad R.K-122.png

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 kami 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 pesawat terbatas pada kerucut.

```
>gp=g();
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0,gp[2]]; dp=gp[3];
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Sekarang kami 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 elips.

```
>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)));
```

Selanjutnya kita menghitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
```

Kami menghubungkan titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang kami membuat 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]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```

Untuk mendapatkan Anaglyph ini, kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

```
>function scene () ...
```

```
global a,u,dd,g,g1,defaultpointsize;
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,gp[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]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

Anda membutuhkan kacamata merah / cyan untuk mengapresiasi efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Contoh 8: Geometri Bumi

Di notebook ini, kami ingin melakukan beberapa komputasi bola. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu dulu.

```
>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 bola).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°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°34.333' E 110°49.683'
S 6°59.050' E 110°24.533'
```

Pertama kita menghitung vektor dari satu bola ke bola lainnya pada bola ideal. Vektor ini adalah [heading, distance] dalam radian. Untuk menghitung jarak di bumi, kita mengalikan dengan jari-jari bumi pada garis lintang 7°.

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-So
```

```
65°20'26.60''
53.8945384608
```

Ini adalah perkiraan yang bagus. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu dekat hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->km //perkiraan jarak FMIPA-Semarang
```

```
88.0114026318
```

Ada fungsi untuk heading, dengan mempertimbangkan bentuk bumi yang elips. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
     65.34°
```

```
1Sudut segitiga melebihi 80° pada bola.
```

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo);
```

```
180°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°)^2->" km^2";  // perkiraan luas segitiga FMIPA-Solo-Semarang
```

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^2
```

Kami juga dapat menambahkan vektor ke posisi. Vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan svector. Untuk menambahkan vektor ke posisi, kami menggunakan saddvector.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(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 429,66 km. Kami mendapatkan perkiraan yang bagus.

```
>esdist(Tugu,Monas)->km // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
431.565659488
```

Judulnya sama dengan yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17'2.85''
```

Namun kita tidak lagi mendapatkan posisi sasaran yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, namun melakukan perkiraan jari-jari bumi di sepanjang lintasan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
S 6°10.500' E 106°48.717'
```

The error is not large, however.

```
>sposprint(Monas),
```

```
S 6°10.500' E 106°48.717'
```

Tentunya kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan tajuk yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh jaraknya, menggunakan heading ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya masih jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
S 6°11.250' E 106°48.372'
      1.529km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada ketinggian yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran dengan garis lintang 30 ?, tetapi jalur yang lebih pendek mulai 10 ? lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
      79.69°
```

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah tujuan kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari jarak total.

```
>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 tajuk yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan heading 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;
>skmprint(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°27.829' E 109°39.196'
S 7°18.219' E 109°17.834'
S 7°8.592' E 108°56.488'
S 6°58.948' E 108°35.157'
S 6°49.289' E 108°13.841'
S 6°39.614' E 107°52.539'
S 6°29.924' E 107°31.251'
S 6°20.219' E 107°9.977'
S 6°10.500' E 106°48.717'
```

Kami menulis sebuah fungsi, yang menggambarkan bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction
```

Now plot everything.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```

Atau gunakan plot3d untuk mendapatkan tampilan anaglyphnya. Ini terlihat sangat bagus dengan kacamata merah / cyan.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```

Latihan Soal

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.

Penyelesaian

```
>o &:= [0,0]; c=circleWithCenter(o,5);
>color(1); setPlotRange(5); plotPoint(o); plotCircle(c);
>A=[-5,0]; plotPoint(A,"A");
>B=[5,0]; plotPoint(B,"B");
>plotSegment(A,B,""):
>c1=circleWithCenter(A,distance(A,o));
>c2=circleWithCenter(B,distance(B,o));
>k=circleCircleIntersections(c1,c);
>l=circleCircleIntersections(c,c2);
>m=circleCircleIntersections(c2,c);
>n=circleCircleIntersections(c,c1);
>r=lineThrough(k,m); s=lineThrough(l,n);
>setPlotRange(8); plotPoint(o); plotCircle(c); plotPoint(A,"A"); plotPoint(B,"B"); plotSeg
>color(4); plotCircle(c1); plotCircle(c2); plotPoint(k); plotPoint(l); plotPoint(m); plotP
>color(5); plotLine(r); plotLine(s):
>color(6); plotSegment(A,k,""); plotSegment(A,n,""); plotSegment(k,l,""); plotSegment(l,B,
```

---

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 tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

Penyelesaian

```
>load geometry;
>setPlotRange(5); P=[1,0]; 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])
```

[]

```
>function y&=-x^2+5*x-4
```

```
                  2
               - x  + 5 x - 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-sisintya 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.

Penyelesaian

```
>setPlotRange(-5,5,-5,5);
>A=[-2,2]; plotPoint(A,"A");
>B=[2,2]; plotPoint(B,"B");
>C=[2,-2]; plotPoint(C,"C");
>D=[-2,-2]; plotPoint(D,"D");
>plotSegment(A,B);
>plotSegment(B,C);
>plotSegment(C,D);
>plotSegment(D,A);
>plotSegment(A,C,"q1");
>plotSegment(B,D,"q2");
```

```
>q1=lineThrough(A,C);
>q2=lineThrough(B,D);
>p=lineIntersection(q1,q2);
>plotLine(q1); plotLine(q2);
>plotPoint(p, "P");
>r=norm(p-projectToLine(p,lineThrough(A,B)))
```

```
2
```

```
>plotCircle(circleWithCenter(p,r),"lingkaran dalam segi-4 ABCD"):
```

---

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

```
>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 hiperbola 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).

Penyelesaian

```
>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):
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

BAB 6

KB PEKAN 10; MENGGUNAKAN EMT UNTUK STATISTIKA

[a4paper,10pt]article eumat

Nama : Akhmad Rizki Khamid
Kelas : Matematika E
NIM : 22305141008

EMT untuk Statistika

Dalam buku catatan ini, kami mendemonstrasikan plot statistik utama, pengujian, dan distribusi di Euler.
Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan beberapa latar belakang untuk memahami detailnya.
Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan standar deviasi yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
>mean(M), dev(M),
```

```
999.9
2.72641400622
```

Kita dapat memplot plot kotak-dan-kumis untuk data. Dalam kasus kami tidak ada outlier.

```
>boxplot(M):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-001.png

Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dan distribusi normal.
Semua fungsi untuk distribusi di Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

$$\text{normaldis(x,m,d)} = \int_{-\infty}^{x} \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} \, dt.$$

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

```
3.07 %
```

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut dalam rentang ukuran yang diberikan.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah plot distribusinya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\/"):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-002.png

Kita bisa memasukkan data mentah tersebut ke dalam sebuah tabel.
Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah orang dalam jangkauan.
Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur header.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T,labc=["from","to","count"])
```

```
   from        to     count
  155.5     159.5        22
  159.5     163.5        71
  163.5     167.5       136
  167.5     171.5       169
  171.5     175.5       139
  175.5     179.5        71
  179.5     183.5        32
  183.5     187.5         8
```

Jika kita membutuhkan nilai rata-rata dan statistik lain dari ukuran, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.
Sumbul "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsion "labc" adalah untuk menentukan header kolom.

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

```
        157.5
        161.5
        165.5
        169.5
        173.5
```

```
177.5
181.5
185.5
```

Tetapi lebih mudah, untuk melipat rentang dengan vektor [1/2.1/2].

```
>M=fold(r,[0.5,0.5])
```

```
[157.5,  161.5,  165.5,  169.5,  173.5,  177.5,  181.5,  185.5]
```

Sekarang kita dapat menghitung rata-rata dan deviasi sampel dengan frekuensi yang diberikan.

```
>{m,d}=meandev(M,v); m, d,
```

```
169.901234568
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai-nilai ke plot batang di atas. Rumus untuk distribusi normal dengan mean m dan standar deviasi d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{\frac{-(x-m)^2}{2d^2}}.$$

Karena nilainya antara 0 dan 1, untuk memplotnya pada bar plot harus dikalikan dengan 4 kali jumlah total data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
>  xmin=min(r),xmax=max(r),thickness=3,add=1):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-003.png

Tabel

Di direktori notebook ini Anda menemukan file dengan tabel. Data tersebut mewakili hasil survei. Berikut adalah empat baris pertama dari file tersebut. Data berasal dari buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat",4);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

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");
```

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);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

```
>load over statistics;
```

Untuk mencetak, kita perlu menentukan set token yang sama. Kami mencetak empat baris pertama saja.

```
>writetable(MT[1:4],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
MT is not a variable!
Error in:
writetable(MT[1:4],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok ...
                  ^
```

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);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Fungsi readtable() sekarang mengembalikan satu set token.

```
>tok
```

```
Variable tok not found!
Error in:
tok ...
   ^
```

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 NAval.

```
>MT[1]
```

```
MT is not a variable!
Error in:
MT[1] ...
     ^
```

Berikut isi tabel dengan angka yang belum diterjemahkan.

```
>writetable(MT,wc=5)
```

```
Variable or function MT not found.
Error in:
writetable(MT,wc=5) ...
             ^
```

Untuk kenyamanan, Anda dapat memasukkan output readtable() ke dalam daftar.

```
>Table={{readtable("table.dat",ctok=ctok)}};
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. Atau menggunakan daftar Tabel.

```
>writetable(Table,ctok=ctok,wc=5);
```

```
Variable or function Table not found.
Error in:
writetable(Table,ctok=ctok,wc=5); ...
                 ^
```

Fungsi tablecol() mengembalikan nilai kolom tabel, melewatkan setiap baris dengan nilai NAN("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

```
Variable or function MT not found.
Error in:
{c,i}=tablecol(MT,[5,6]); ...
                  ^
```

Kita dapat 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)
```

```
Variable or function i not found.
Error in:
j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok) ...
                           ^
```

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Tabel dalam kasus ini.

```
>MT=Table[1];
```

```
Table is not a variable!
Error in:
MT=Table[1]; ...
           ^
```

Tentu saja, kita juga dapat menggunakannya untuk menentukan nilai rata-rata kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT,6))
```

```
Variable or function MT not found.
Error in:
mean(tablecol(MT,6)) ...
                 ^
```

Fungsi getstatistics() mengembalikan elemen dalam vektor, dan jumlahnya. Kami menerapkannya pada nilai "m" dan "f" di kolom kedua tabel kami.

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

```
Variable or function MT not found.
Error in:
{xu,count}=getstatistics(tablecol(MT,2)); xu, count, ...
                                    ^
```

Kami dapat mencetak hasilnya dalam tabel baru.

```
>writetable(count',labr=tok[xu])
```

```
Variable count not found!
Error in:
writetable(count',labr=tok[xu]) ...
                     ^
```

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"])
```

```
Variable or function tok not found.
Error in:
v:=indexof(tok,["g","vg"]) ...
                  ^
```

Sekarang kita dapat memilih baris tabel, yang memiliki salah satu nilai dalam v di baris ke-5.

```
>MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

```
Variable or function MT not found.
Error in:
MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5); ...
                         ^
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstrak dan diurutkan di kolom ke-5.

```
>writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);
```

```
Variable or function i not found.
Error in:
writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7); ...
                 ^
```

Untuk statistik berikutnya, kami ingin menghubungkan dua kolom tabel. Jadi kami mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

```
>i=sortedrows(MT,[2,4]);  ...
>  writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

```
Variable or function MT not found.
Error in:
i=sortedrows(MT,[2,4]);    writetable(tablecol(MT[i],[2,4])',c ...
                ^
```

Dengan getstatistics(), kita juga dapat menghubungkan hitungan 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])
```

```
Variable or function MT not found.
Error in:
MT24=tablecol(MT,[2,4]); {xu1,xu2,count}=getstatistics(MT24[1] ...
                ^
```

Sebuah tabel dapat ditulis ke file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

```
Variable or function count not found.
Error in:
filename="test.dat"; writetable(count,labr=tok[xu1],labc=tok[x ...
                                     ^
```

Kemudian kita bisa membaca tabel dari file.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
>writetable(MT2,labr=hdr,labc=hd)
```

```
Could not open the file
test.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Dan hapus filenya.

```
>fileremove(filename);
```

Distribusi

Dengan plot2d, ada metode yang sangat mudah untuk memplot distribusi data eksperimen.

```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="\/"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-004.png

Harap dicatat perbedaan antara plot batang (sampel) dan kurva normal (distribusi nyata). Masukkan kembali tiga perintah untuk melihat hasil pengambilan sampel lainnya.

Berikut adalah perbandingan 10 simulasi 1000 nilai terdistribusi normal menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, dan outlier.

```
>p=normal(10,1000); boxplot(p):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-005.png

Untuk menghasilkan bilangan bulat acak, Euler memiliki intrarandom. Mari kita simulasikan lemparan dadu dan plot distribusinya.
Kami menggunakan fungsi getmultiplicities(v,x), yang menghitung seberapa sering elemen v muncul di x. Kemudian kita plot hasilnya menggunakan columnplot().

```
>k=intrandom(1,6000,6);  ...
>columnsplot(getmultiplicities(1:6,k));  ...
>ygrid(1000,color=red):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-006.png

Sementara intrandom(n,m,k) mengembalikan bilangan bulat terdistribusi seragam dari 1 ke k, dimungkinkan untuk menggunakan distribusi bilangan bulat lainnya dengan randpint().
Dalam contoh berikut, probabilitas untuk 1,2,3 berturut-turut adalah 0,4,0,1,0,5.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
  [378,  102,  520]
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Coba lihat referensinya.
Misalnya, kami mencoba distribusi eksponensial. Sebuah variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

with parameter

$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-007.png

Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan kebalikannya.

```
>plot2d("normaldis",-4,4):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-008.png

Berikut ini adalah salah satu cara untuk memplot kuantil.

```
>plot2d("qnormal(x,1,1.5)",-4,6);  ...
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-009.png

$$\text{normaldis(x,m,d)} = \int_{-\infty}^{x} \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2}\ dt.$$

Peluang berada di area hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

```
0.248662156979
```

Ini dapat dihitung secara numerik dengan integral berikut.

$$\int_{2}^{5} \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2}\ dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

```
0.248662156979
```

Mari kita bandingkan distribusi binomial dengan distribusi normal mean dan deviasi yang sama. Fungsi invbindis() memecahkan interpolasi linier antara nilai integer.

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219
526.007419394
```

Fungsi qdis() adalah densitas dari distribusi chi-kuadrat. Seperti biasa, evolusi vektor ke fungsi ini. Dengan demikian kita mendapatkan plot semua distribusi chi-kuadrat dengan derajat 5 sampai 30 dengan mudah dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50)')",0,50):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-010.png

Euler memiliki fungsi yang akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.
Penamaan mencoba untuk konsisten. Misalnya.,
- distribusi chi-kuadrat adalah chidis(),
- fungsi kebalikannya adalah invchidis(),
- kepadatannya adalah qchidis().

Komplemen dari distribusi (ekor atas) adalah chicdis().

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259
0.527633447259
```

Distribusi Diskrit

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.
Pertama kita atur fungsi distribusinya.

```
>wd = 0|((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0,  0.165,  0.335,  0.5,  0.666667,  0.833333,  1]
```

Artinya dengan probabilitas wd[i+1]-wd[i] kita menghasilkan nilai acak i.
Ini adalah distribusi yang hampir seragam. Mari kita mendefinisikan generator nomor acak untuk ini. Fungsi find(v,x) menemukan nilai x dalam vektor v. Fungsi ini juga berfungsi untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya sangat halus sehingga kita hanya melihatnya dengan sangat banyak iterasi.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-011.png

Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai 1...K dalam v. Kami 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 binomialsum(), yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

```
>bindis(410,1000,0.4)
```

```
0.751401349654
```

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat 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,  0.441212]
```

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Tetapi untuk n besar, penjumlahan langsung tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
0.751401349655
```

Omong-omong, invbinsum() menghitung kebalikan dari binomialsum().

```
>invbindis(0.75,1000,0.4)
```

```
409.932733047
```

Di Bridge, kami mengasumsikan 5 kartu yang beredar (dari 52) di dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (misalnya 0:5, 1:4, 4:1 atau 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

```
0.321739130435
```

Ada juga simulasi distribusi multinomial.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

```
          381           100           519
          376            91           533
          417            80           503
          440            94           466
          406           112           482
          408            94           498
          395           107           498
          399            96           505
          428            87           485
          400            99           501
```

Merencanakan Data

Untuk plot data, kami mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam 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];
```

Untuk para pihak, kami menggunakan serangkaian nama.

```
>P:=["CDU/CSU","SPD","FDP","Gr","Li"];
```

Mari kita mencetak persentase dengan baik.
Pertama kita ekstrak kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi. kolom 1 adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama sebagai tajuk kolom, dan tahun sebagai tajuk untuk baris. Lebar default untuk kolom adalah wc=10, tetapi kami lebih memilih output yang lebih padat. Kolom akan diperluas untuk label kolom, jika perlu.

```
>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
```

Perkalian matriks berikut mengekstrak jumlah persentase dua partai besar yang menunjukkan bahwa partai-partai kecil telah memperoleh rekaman di parlemen hingga 2009.

```
>BT1:=(BT.[1;1;0;0;0])'*100
```

```
 [84.29,  81.25,  81.1659,  82.7529,  72.9642,  61.8971,  79.8732]
```

Ada juga plot statistik sederhana. Kami menggunakannya untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-012.png

Tentukan beberapa warna untuk masing-masing pihak.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

Sekarang kita dapat memplot hasil pemilu 2009 dan perubahannya menjadi satu plot menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

```
>figure(2,1);  ...
>figure(1); columnsplot(BW[6,3:7],P,color=CP); ...
>figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP);  ...
>figure(0):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-013.png

Plot data menggabungkan deretan data statistik dalam satu plot.

```
>J:=BW[,1]'; DP:=BW[,3:7]'; ...
>dataplot(YT,BT',color=CP);  ...
>labelbox(P,colors=CP,styles="[]",>points,w=0.2,x=0.3,y=0.4):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-014.png

Sebuah kolom plot 3D menunjukkan baris data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. angle adalah sudut pandang.

```
>columnsplot3d(BT,scols=P,srows=YT, ...
>  angle=30°,ccols=CP):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-015.png

Representasi lain adalah plot mosaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjangnya label CDU/CSU, kami mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller);  ...
>mosaicplot(BT',srows=YT,scols=P,color=CP,style="#"); ...
>shrinkwindow():
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-016.png

Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning membentuk koalisi, kami menyusun ulang elemen-elemennya.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-017.png

Berikut adalah jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-018.png

Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls dari data acak, terdistribusi secara merata di [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-019.png

Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-020.png

Fungsi columnplot() lebih mudah digunakan, karena hanya membutuhkan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kita sudah mendemonstrasikannya dalam tutorial ini.
Ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan menyusun statistik.

```
>v=strtochar("the quick brown fox jumps over the lazy dog"); ...
>w=ascii("a"):ascii("z"); x=getmultiplicities(w,v); ...
>cw=[]; for k=w; cw=cw|char(k); end; ...
>columnsplot(x,lab=cw,width=0.05):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-021.png

Dimungkinkan juga untuk mengatur sumbu secara manual.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
>yaxis(0,0:0.1:1,style="->",>left); xaxis(0,style="."); ...
>label("p",0,0.25), label("i",11,0); ...
>textbox(["Binomial distribution","with p=0.4"]):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-022.png

Berikut ini adalah cara untuk memplot frekuensi bilangan dalam sebuah vektor.
Kami membuat vektor bilangan bulat bilangan acak 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

```
[8,  5,  8,  8,  6,  8,  8,  3,  5,  5]
```

Kemudian ekstrak nomor unik di v.

```
>vu:=unique(v)
```

```
[3,  5,  6,  8]
```

Dan plot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-023.png

Kami ingin menunjukkan fungsi untuk distribusi nilai empiris.

```
>x=normal(1,20);
```

Fungsi empdist(x,vs) membutuhkan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

```
>xs=sort(x);
```

Kemudian kami memplot distribusi empiris dan beberapa batang kepadatan menjadi satu plot. Alih-alih plot bar untuk distribusi, kami menggunakan plot gigi gergaji kali ini.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar));  ...
>figure(0):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-024.png

Plot pencar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa X dan X+Y jelas berkorelasi positif.

```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-025.png

Seringkali, kita ingin membandingkan dua sampel dari distribusi yang berbeda. Ini dapat dilakukan dengan plot kuantil-kuantil.
Untuk pengujian, kami mencoba distribusi student-t dan distribusi eksponensial.

```
>x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ...
>plot2d("x",r=6,style="--",yl="normal",xl="student-t",>vertical); ...
>plot2d(sort(x),sort(y),>points,color=red,style="x",>add):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-026.png

Plot dengan jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.
Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat memperluas yang lebih kecil atau mengecilkan yang lebih besar. Fungsi berikut baik untuk keduanya. Dibutuhkan nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari kita bandingkan dua distribusi yang sama.

```
>x=random(1000); y=random(400); ...
>plot2d("x",0,1,style="--"); ...
>plot2d(sort(medianexpand(x,400)),sort(y),>points,color=red,style="x",>add):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-027.png

Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi fit.
Sebagai permulaan, kami menemukan garis regresi untuk data univariat dengan polifit(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
```

Kami ingin membandingkan non-weighted dan weighted fit. Pertama, koefisien kecocokan linier.

```
>p=polyfit(x,y,1)
```

```
  [0.733333,  0.812121]
```

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]
```

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):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-028.png

Sebagai contoh lain kita membaca survei siswa, usia mereka, usia orang tua mereka dan jumlah saudara kandung dari sebuah file.
Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel tok2 untuk mengatur terjemahan yang tepat daripada membiarkan readtable() mengumpulkan terjemahan.

```
>{MS,hd}:=readtable("table1.dat",tok2:=["m","f"]);  ...
>writetable(MS,labc=hd,tok2:=["m","f"]);
```

```
Could not open the file
table1.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Bagaimana usia bergantung satu sama lain? Kesan pertama datang dari scatterplot berpasangan.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

```
Variable or function MS not found.
Error in:
scatterplots(tablecol(MS,3:5),hd[3:5]): ...
                        ^
```

Jelas bahwa usia ayah dan ibu bergantung satu sama lain. Mari kita tentukan dan plot garis regresinya.

```
>cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
```

```
MS is not a variable!
Error in:
cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1) ...
            ^
```

Ini jelas model yang salah. Garis regresinya adalah s=17+0,74t, di mana t adalah usia ibu dan s usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.
Sebaliknya, kami menduga fungsi seperti s=a+t. Maka a adalah mean dari s-t. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

```
cs is not a variable!
Error in:
da:=mean(cs[2]-cs[1]) ...
              ^
```

Mari kita plot ini menjadi satu plot pencar.

```
>plot2d(cs[1],cs[2],>points);  ...
>plot2d("evalpoly(x,ps)",color=red,style=".",>add);  ...
>plot2d("x+da",color=blue,>add):
```

```
cs is not a variable!
Error in:
plot2d(cs[1],cs[2],>points);  plot2d("evalpoly(x,ps)",color=re ...
            ^
```

Berikut adalah plot kotak dari dua zaman. Ini hanya menunjukkan, bahwa usianya berbeda.

```
>boxplot(cs,["mothers","fathers"]):
```

```
Variable or function cs not found.
Error in:
boxplot(cs,["mothers","fathers"]): ...
          ^
```

Sangat menarik bahwa perbedaan median tidak sebesar perbedaan rata-rata.

```
>median(cs[2])-median(cs[1])
```

```
cs is not a variable!
Error in:
median(cs[2])-median(cs[1]) ...
            ^
```

Koefisien korelasi menunjukkan korelasi positif.

```
>correl(cs[1],cs[2])
```

```
cs is not a variable!
Error in:
correl(cs[1],cs[2]) ...
            ^
```

Korelasi peringkat adalah ukuran untuk urutan yang sama di kedua vektor. Ini juga cukup positif.

```
>rankcorrel(cs[1],cs[2])
```

```
cs is not a variable!
Error in:
rankcorrel(cs[1],cs[2]) ...
                ^
```

Membuat Fungsi baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi-fungsi baru. Misalnya, kita mendefinisikan fungsi skewness.

$$\text{sk}(x) = \frac{\sqrt{n}\sum_i (x_i - m)^3}{\left(\sum_i (x_i - m)^2\right)^{3/2}}$$

dimana m adalah mean dari x.

```
>function skew (x:vector) ...
```

```
m=mean(x);
return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2);
endfunction
```

Seperti yang Anda lihat, kita dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```
>data=normal(20); skew(normal(10))
```

```
-0.198710316203
```

Berikut adalah fungsi lain, yang disebut koefisien skewness Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

```
-0.0801873249135
```

Simulasi Monte Carlo

Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Ini adalah satu lagi, yang mensimulasikan 1000 kali 3 lemparan dadu, dan meminta distribusi jumlah.

```
>ds:=sum(intrandom(1000,3,6))'; fs=getmultiplicities(3:18,ds)
```

```
[5,  17,  35,  44,  75,  97,  114,  116,  143,  116,  104,  53,  40,
22,  13,  6]
```

kita bisa membuat plot ini sekarang

```
>columnsplot(fs,lab=3:18):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-029.png

Untuk menentukan distribusi yang diharapkan tidak begitu mudah. Kami menggunakan rekursi lanjutan untuk ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat direpresentasikan sebagai jumlah n bilangan dalam rentang 1 sampai m. Ia 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
```

Berikut adalah hasil dari tiga lemparan dadu.

```
>cw=countways(3:18,3,6)
```

```
  [1,   3,   6,   10,   15,   21,   25,   27,   27,   25,   21,   15,   10,   6,   3,
  1]
```

Kami menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-030.png

Untuk simulasi lain, simpangan nilai rata-rata dari n 0-1-variabel acak terdistribusi normal adalah 1/sqrt(n).

```
>longformat; 1/sqrt(10)
```

```
  0.316227766017
```

Mari kita periksa ini dengan simulasi. Kami memproduksi 10000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M)')
```

```
0.319493614817
```

```
>plot2d(mean(M)',>distribution):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-031.png

Median 10 0-1-bilangan acak terdistribusi normal memiliki simpangan yang lebih besar.

```
>dev(median(M)')
```

```
0.374460271535
```

Karena kita dapat dengan mudah menghasilkan jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan rata-rata dari langkah ke-n dari proses ini bersama dengan nilai yang diharapkan dalam warna merah.

```
>n=1000; m=1000; M=cumsum(normal(n,m)/sqrt(m)); ...
>t=(1:n)/n; figure(2,1); ...
>figure(1); plot2d(t,mean(M')'); plot2d(t,0,color=red,>add); ...
>figure(2); plot2d(t,dev(M')'); plot2d(t,sqrt(t),color=red,>add); ...
>figure(0):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-032.png

Uji

Uji adalah alat penting dalam statistik. Di Euler, banyak tes diimplementasikan. Semua tes ini mengembalikan kesalahan yang kami terima jika kami menolak hipotesis nol.
Sebagai contoh, kami menguji lemparan dadu untuk distribusi seragam. 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
```

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)
```

```
0.526
```

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak distribusi seragam. 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)')
```

```
0.528028118442
```

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

```
0.0218365848476
```

Fungsi ttest() membutuhkan nilai rata-rata, simpangan, jumlah data, dan nilai rata-rata yang akan diuji.
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))
```

```
0.38722000942
```

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)
```

```
5.60009101758e-07
```

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)
```

```
3.28
```

Kami sekarang membandingkan jumlah satu dengan distribusi binomial. Pertama kita plot distribusi yang.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\/"):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-033.png

```
>t=count(R,21);
```

Kemudian kami 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)
```

```
0.53921579764
```

Contoh berikut berisi hasil dua kelompok orang (laki-laki dan perempuan, katakanlah) memberikan suara untuk satu dari enam partai.

```
>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
```

Kami ingin menguji independensi suara dari jenis kelamin. Tes tabel chi^2 melakukan ini. Akibatnya terlalu besar untuk menolak kemerdekaan. Jadi kita tidak bisa mengatakan, jika voting tergantung pada jenis kelamin dari data ini.

```
>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 pemungutan suara tidak tergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Uji Lainnya

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
```

Ini berarti, kami menolak hipotesis nilai rata-rata yang sama. Kami melakukan ini dengan probabilitas kesalahan 1,3%.
Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata berbeda menguji median sampel bersatu.

```
>a=[56,66,68,49,61,53,45,58,54];
>b=[72,81,51,73,69,78,59,67,65,71,68,71];
>mediantest(a,b)
```

```
0.0241724220052
```

Tes lain tentang kesetaraan adalah tes peringkat. Ini jauh lebih tajam daripada tes median.

```
>ranktest(a,b)
```

```
0.00199969612469
```

Dalam contoh berikut, kedua distribusi memiliki mean yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

```
0.129608141484
```

Sekarang mari kita coba mensimulasikan dua perlakuan a dan b yang diterapkan pada orang yang berbeda.

```
>a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];
>b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];
```

Tes signum memutuskan, jika a lebih baik dari b.

```
>signtest(a,b)
```

```
0.0546875
```

Ini terlalu banyak kesalahan. Kita tidak dapat menolak bahwa a sama baiknya dengan b.
Tes Wilcoxon lebih tajam dari tes ini, tetapi bergantung pada nilai kuantitatif perbedaan.

```
>wilcoxon(a,b)
```

```
0.0296680599405
```

Mari kita coba dua tes lagi menggunakan seri yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

```
0.0068706451766
```

```
>wilcoxon(normal(1,20),normal(1,20))
```

```
0.275145971064
```

Nomor Acak

Berikut ini adalah pengujian untuk pembangkit bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan masalah.
Pertama kita menghasilkan sepuluh juta angka acak di [0,1].

```
>n:=10000000; r:=random(1,n);
```

Selanjutnya kita hitung jarak antara dua bilangan kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Akhirnya, kami memplot berapa kali, setiap jarak terjadi, dan membandingkan dengan nilai yang diharapkan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
>  plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-034.png

Hapus datanya.

```
>remvalue n;
```

Pengantar untuk Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan paket numerik dan sistem aljabar komputer.

Notebook ini cocok untuk Anda yang terbiasa dengan R, tetapi perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.

Perhatikan bahwa ini adalah pekerjaan yang sedang berjalan.

Sintaks Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Di EMT, perbedaan utama adalah bahwa : operator dapat mengambil ukuran langkah. Selain itu, ia memiliki daya ikat yang rendah.

```
>n=10; 0:n/20:n-1
```

```
[0,  0.5,  1,  1.5,  2,  2.5,  3,  3.5,  4,  4.5,  5,  5.5,  6,  6.5,
7,  7.5,  8,  8.5,  9]
```

Fungsi c() tidak ada. Dimungkinkan untuk menggunakan vektor untuk menggabungkan sesuatu.

Contoh berikut, seperti banyak contoh lainnya, dari "Interoduction to R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti jalannya dalam tutorial ini.

```
>x=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
```

```
[10.4,  5.6,  3.1,  6.4,  21.7,  0,  10.4,  5.6,  3.1,  6.4,  21.7]
```

Operator titik dua dengan ukuran langkah EMT diganti dengan fungsi seq() di R. Kita bisa menulis fungsi ini di EMT.

```
>function seq(a,b,c) := a:b:c; ...
>seq(0,-0.1,-1)
```

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

Fungsi rep() dari R tidak ada di EMT. Untuk input vektor, dapat ditulis sebagai berikut.

```
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...
>rep(x,2)
```

```
[10.4,  5.6,  3.1,  6.4,  21.7,  10.4,  5.6,  3.1,  6.4,  21.7]
```

Perhatikan bahwa "=" atau ":=" digunakan untuk tugas. Operator "->" digunakan untuk unit di EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

Operator "<-" untuk penugasan tetap menyesatkan, dan bukan ide yang baik untuk R. Berikut ini akan membandingkan a dan -4 di EMT.

```
>a=2; a<-4
```

```
0
```

Di R, "a<-4<3" berfungsi, tetapi "a<-4<-3" tidak. Saya juga memiliki ambiguitas serupa di EMT, tetapi mencoba menghilangkannya perlahan-lahan.
EMT dan R memiliki vektor bertipe boolean. Namun di EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Di R, nilai true dan false dapat digunakan dalam aritmatika biasa seperti di EMT.

```
>x<5, %*x
```

```
[0,   0,   1,   0,   0]
[0,   0,   3.1,   0,   0]
```

EMT melempar kesalahan atau menghasilkan NAN tergantung pada tanda "kesalahan".

```
>errors off; 0/0, isNAN(sqrt(-1)), errors on;
```

```
NAN
1
```

String sama di R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.
Di R ada paket untuk Unicode. Di EMT, sebuah string dapat berupa string Unicode. String unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u"..." dapat berisi entitas HTML.

```
>u"&#169; Ren&eacut; Grothmann"
```

```
© René Grothmann
```

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar di sistem Anda sebagai A dengan titik dan garis di atasnya. Itu tergantung pada font yang Anda gunakan.

```
>chartoutf([480])
```

Penggabungan string dilakukan dengan "+" atau "|". Ini dapat mencakup angka, yang akan dicetak dalam format saat ini.

```
>"pi = "+pi
```

```
pi = 3.14159265359
```

Pengindeksan

Sebagian besar waktu, ini akan berfungsi seperti pada R.
Tetapi EMT akan menginterpretasikan indeks negatif dari belakang vektor, sedangkan R menginterpretasikan x[n] sebagai x tanpa elemen ke-n.

```
>x, x[1:3], x[-2]
```

```
[10.4,   5.6,   3.1,   6.4,   21.7]
[10.4,   5.6,   3.1]
6.4
```

Perilaku R dapat dicapai dalam EMT dengan drop().

```
>drop(x,2)
```

```
[10.4,   3.1,   6.4,   21.7]
```

Vektor logis tidak diperlakukan secara berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu mengekstrak elemen bukan nol terlebih dahulu di EMT.

```
>x, x>5, x[nonzeros(x>5)]
```

```
[10.4,   5.6,   3.1,   6.4,   21.7]
[1,   1,   0,   1,   1]
[10.4,   5.6,   6.4,   21.7]
```

Sama seperti di R, vektor indeks dapat berisi pengulangan.

```
>x[[1,2,2,1]]
```

```
[10.4,   5.6,   5.6,   10.4]
```

Tetapi nama untuk indeks tidak dimungkinkan di EMT. Untuk paket statistik, ini mungkin sering diperlukan untuk memudahkan akses ke elemen vektor.
Untuk meniru perilaku ini, kita dapat mendefinisikan fungsi sebagai berikut.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...
>s=["first","second","third","fourth"]; sel(x,["first","third"],s)
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
             ^
[10.4,   3.1]
```

Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelas, di R ada vektor yang tumbuh. Anda dapat mengatur vektor numerik kosong v dan menetapkan nilai ke elemen v[17]. Ini tidak mungkin di EMT.
Berikut ini agak tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT sekarang akan membuat vektor dengan v dan i ditambahkan pada tumpukan dan menyalin vektor itu kembali ke variabel global v.
Semakin efisien pra-mendefinisikan vektor.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah jenis tanggal di EMT, Anda dapat menggunakan fungsi seperti complex().

```
>complex(1:4)
```

```
[ 1+0i ,  2+0i ,  3+0i ,  4+0i  ]
```

Konversi ke string hanya dimungkinkan untuk tipe data dasar. Format saat ini digunakan untuk rangkaian string sederhana. Tetapi ada fungsi seperti print() atau frac().
Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
>function tostr (v) ...
```

```
s="[";
loop 1 to length(v);
   s=s+print(v[#],2,0);
   if #<length(v) then s=s+","; endif;
end;
return s+"]";
endfunction
```

```
>tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk komunikasi dengan Maxima, terdapat fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk output.

```
>convertmxm(1:10)
```

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

Untuk Latex perintah tex dapat digunakan untuk mendapatkan perintah Latex.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

Faktor dan Tabel

Dalam pengantar R ada contoh dengan apa yang disebut faktor.
Berikut ini adalah daftar wilayah dari 30 negara bagian.

```
>austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ...
>"qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", ...
>"sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", ...
>"sa", "act", "nsw", "vic", "vic", "act"];
```

Asumsikan, kita memiliki pendapatan yang sesuai di setiap negara bagian.

```
>incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ...
>61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ...
>59, 46, 58, 43];
```

Sekarang, kami ingin menghitung rata-rata pendapatan di wilayah tersebut. Menjadi program statistik, R memiliki factor() dan tappy() untuk ini.
EMT dapat melakukannya dengan menemukan indeks wilayah dalam daftar wilayah yang unik.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
```

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

Pada titik itu, kita dapat menulis fungsi loop kita sendiri untuk melakukan sesuatu hanya untuk satu faktor.
Atau kita bisa meniru fungsi tapply() dengan cara berikut.

```
>function map tappl (i; f$:call, cat, x) ...
```

```
u=sort(unique(cat));
f=indexof(u,cat);
return f$(x[nonzeros(f==indexof(u,i))]);
endfunction
```

Ini agak tidak efisien, karena menghitung wilayah unik untuk setiap i, tetapi berhasil.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5,  57.3333333333,  55.5,  53.6,  55,  60.5,  56,  52.25]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5,  57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.
Jadi kita bisa mencetak rata-rata pendapatan negara di wilayah dengan cara yang bersahabat.

```
>writetable(tappl(auterr,"mean",austates,incomes),labc=auterr,wc=7)
```

```
    act    nsw     nt    qld     sa    tas    vic     wa
   44.5  57.33   55.5   53.6     55   60.5     56  52.25
```

Kita juga dapat mencoba meniru perilaku R sepenuhnya.
Faktor-faktor tersebut harus dengan jelas disimpan dalam kumpulan dengan jenis dan kategori (negara bagian dan teritori dalam contoh kami). Untuk EMT, kami menambahkan indeks yang telah dihitung sebelumnya.

```
>function makef (t) ...
```

```
## Factor data
## Returns a collection with data t, unique data, indices.
## See: tapply
u=sort(unique(t));
return {{t,u,indexofsorted(u,t)}};
endfunction
```

```
>statef=makef(austates);
```

Sekarang elemen ketiga dari koleksi akan berisi indeks.

```
>statef[3]
```

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

Sekarang kita bisa meniru tapply() dengan cara berikut. Ini akan mengembalikan tabel sebagai kumpulan data tabel dan judul kolom.

```
>function tapply (t:vector,tf,f$:call) ...
```

```
## Makes a table of data and factors
## tf : output of makef()
## See: makef
uf=tf[2]; f=tf[3]; x=zeros(length(uf));
for i=1 to length(uf);
   ind=nonzeros(f==i);
   if length(ind)==0 then x[i]=NAN;
   else x[i]=f$(t[ind]);
   endif;
end;
return {{x,uf}};
endfunction
```

Kami tidak menambahkan banyak jenis pengecekan di sini. Satu-satunya tindakan pencegahan menyangkut kategori (faktor) tanpa data. Tetapi orang harus memeriksa panjang t yang benar dan kebenaran koleksi tf.
Tabel ini dapat dicetak sebagai tabel dengan writetable().

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

```
    act    nsw     nt    qld     sa    tas    vic     wa
   44.5  57.33   55.5   53.6     55   60.5     56  52.25
```

Array

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau pustaka C untuk ini.
R memiliki lebih dari dua dimensi. Dalam R array adalah vektor dengan bidang dimensi.
Dalam EMT, vektor adalah matriks dengan satu baris. Itu dapat dibuat menjadi matriks dengan redim().

```
>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
```

Ekstraksi baris dan kolom, atau sub-matriks, sangat mirip dengan R.

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

```
          2           3
          7           8
         12          13
         17          18
```

Namun, dalam R dimungkinkan untuk menetapkan daftar indeks spesifik dari vektor ke suatu nilai. Hal yang sama dimungkinkan di EMT hanya dengan loop.

```
>function setmatrixvalue (M, i, j, v) ...
```

```
  loop 1 to max(length(i),length(j),length(v))
     M[i{#},j{#}] = v{#};
  end;
  endfunction
```

Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalinnya ke dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

```
          1           2           0           4           5
          6           0           8           9          10
          0          12          13          14          15
         16          17          18          19          20
```

Perkalian luar dalam EMT hanya dapat dilakukan antar vektor. Ini otomatis karena bahasa matriks. Satu vektor harus menjadi vektor kolom dan yang lainnya vektor baris.

```
>(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
```

Dalam pengantar PDF untuk R ada sebuah contoh, yang menghitung distribusi ab-cd untuk a,b,c,d yang dipilih dari 0 hingga n secara acak. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan table() di atasnya.
Tentu saja, ini dapat dicapai dengan loop. Tapi loop tidak efektif di EMT atau R. Di EMT, kita bisa menulis loop di C dan itu akan menjadi solusi tercepat.
Tapi kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
>u=sort(unique(q)); f=getmultiplicities(u,q); ...
>statplot(u,f,"h"):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-035.png

Selain multiplisitas yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)
```

```
[0,  23,  132,  316,  602,  801,  333,  141,  53,  0]
```

Cara paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

```
>plot2d(q,distribution=11):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-036.png

Tetapi juga dimungkinkan untuk menghitung sebelumnya hitungan dalam interval yang dipilih sebelumnya. Tentu saja, berikut ini menggunakan getfrequencies() secara internal.
Karena fungsi histo() mengembalikan frekuensi, kita perlu menskalakannya sehingga integral di bawah grafik batang adalah 1.

```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ...
>plot2d(x,y,>bar,style="/"):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-037.png

Daftar

EMT memiliki dua macam daftar. Salah satunya adalah daftar global yang dapat diubah, dan yang lainnya adalah jenis daftar yang tidak dapat diubah. Kami tidak peduli dengan daftar global di sini.
Jenis daftar yang tidak dapat diubah disebut koleksi di EMT. Itu berperilaku seperti struktur di C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

```
>L={{"Fred","Flintstone",40,[1990,1992]}}
```

```
Fred
Flintstone
40
[1990,  1992]
```

Saat ini elemen tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Mereka diakses dengan angka.

```
>(L[4])[2]
```

```
1992
```

File Input dan Output (Membaca dan Menulis Data)

Anda akan sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberitahu Anda tentang banyak cara untuk mencapai ini. Fungsi sederhana adalah writematrix() dan readmatrix().
Mari kita tunjukkan cara membaca dan menulis vektor real ke file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49815
0.28037
```

Untuk menulis data ke file, kita menggunakan fungsi writematrix().
Karena pengenalan ini kemungkinan besar berada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk notebook sendiri, ini tidak perlu, karena file data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita menulis vektor kolom a’ ke file. Ini menghasilkan satu nomor di setiap baris file.

```
>writematrix(a',filename);
```

Untuk membaca data, kita gunakan readmatrix().

```
>a=readmatrix(filename)';
```

dan hapus file ini.

```
>fileremove(filename);
>mean(a), dev(a),
```

```
0.49815
0.28037
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.
Misalnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File "test.csv" berikut akan muncul di folder cuurent Anda.

```
>filename="test.csv"; ...
>writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka file ini dengan Excel Indonesia secara langsung.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut ini.

```
>s1:="f m m f m m m f f f m m f";  ...
>s2:="f f f m m f f";
```

Untuk tokenize ini, kita mendefinisikan vektor token.

```
>tok:=["f","m"]
```

```
f
m
```

Kemudian kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...
>  getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan header token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

```
                f       m
       1        6       7
       2        5       2
```

Untuk statika, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...
>writeln("A,B,C"); writematrix(random(3,3)); ...
>close();
```

Filenya terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C
0.7003664386138074,0.1875530821001213,0.3262339279660414
0.5926249243193858,0.1522927283984059,0.368140583062521
0.8065535209872989,0.7265910840408142,0.7332619844597152
```

Fungsi readtable() dalam bentuknya yang paling sederhana dapat membaca ini dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Koleksi ini dapat dicetak dengan writetable() ke notebook, atau ke file.

```
>writetable(L,wc=10,dc=5)
```

```
         A         B         C
   0.70037   0.18755   0.32623
   0.59262   0.15229   0.36814
   0.80655   0.72659   0.73326
```

Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean() dalam EMT menghitung nilai rata-rata dari baris matriks.

```
>mean(L[1])
```

```
    0.40472
    0.37102
    0.75547
```

File CSV

Pertama, mari kita menulis matriks ke dalam file. Untuk output, kami membuat file di direktori kerja saat ini.

```
>file="test.csv";  ...
>M=random(3,3); writematrix(M,file);
```

Berikut adalah isi dari file ini.

```
>printfile(file)
```

```
  0.8221197733097619,0.821531098722547,0.7771240608094004
  0.8482947121863489,0.3237767724883862,0.6501422353377985
  0.1482301827518109,0.3297459716109594,0.6261901074210923
```

CVS ini dapat dibuka pada sistem bahasa Inggris ke 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)
```

```
    0.82212    0.82153    0.77712
    0.84829    0.32378    0.65014
    0.14823    0.32975    0.62619
```

Dimungkinkan untuk menulis beberapa matriks ke satu file. Perintah open() dapat membuka file untuk ditulis dengan parameter "w". Standarnya adalah "r" untuk membaca.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh garis kosong. Untuk membaca matriks, buka file dan panggil readmatrix() beberapa kali.

```
>open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

```
          1          0          0
          0          1          0
          0          0          1
```

Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai yang 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 adalah contoh.

```
>printfile("excel-data.csv")
```

```
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

Seperti yang Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubah ini di pengaturan sistem atau di Excel, tetapi tidak perlu membaca matriks ke dalam EMT.
Cara termudah untuk membaca ini ke dalam Euler adalah readmatrix(). Semua koma diganti dengan titik dengan parameter >comma. Untuk CSV bahasa Inggris, cukup abaikan parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

```
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
readmatrix:
    if filename<>"" then open(filename,"r"); endif;
```

Mari kita plot ini.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green]'):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-038.png

Ada cara yang lebih mendasar untuk membaca data dari file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari baris data. Secara default, ia mengharapkan titik desimal. Tapi itu juga bisa menggunakan koma desimal, jika Anda memanggil setdecimaldot(",") sebelum Anda menggunakan fungsi ini.
Fungsi berikut adalah contoh untuk ini. Ini akan berhenti di akhir file atau baris kosong.

```
>function myload (file) ...
```

```
open(file);
M=[];
repeat
   until eof();
   v=getvectorline(3);
   if length(v)>0 then M=M_v; else break; endif;
end;
return M;
close(file);
endfunction
```

```
>myload(file)
```

```
    0.82212          0    0.82153          0    0.77712
    0.84829          0    0.32378          0    0.65014
    0.14823          0    0.32975          0    0.62619
```

Dimungkinkan juga untuk membaca semua angka dalam file itu dengan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
    0.82212         0   0.82153
          0   0.77712   0.84829
          0   0.32378         0
```

Jadi sangat mudah untuk menyimpan vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

```
  0.50303
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
  0.50303
```

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kami menulis tabel dengan header baris dan kolom ke file.

```
>file="test.tab"; M=random(3,3);  ...
>open(file,"w");  ...
>writetable(M,separator=",",labc=["one","two","three"]);  ...
>close(); ...
>printfile(file)
```

```
  one,two,three
        0.09,      0.39,      0.86
        0.39,      0.86,      0.71
         0.2,      0.02,      0.83
```

Ini dapat diimpor ke Excel.
Untuk membaca file dalam EMT, kami menggunakan readtable().

```
>{M,headings}=readtable(file,>clabs); ...
>writetable(M,labc=headings)
```

```
         one       two     three
        0.09      0.39      0.86
        0.39      0.86      0.71
         0.2      0.02      0.83
```

Menganalisis Garis

Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki garis dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03,Tue,1'114.05
```

Pertama kita dapat menandai garis.

```
>vt=strtokens(line)
```

```
2020-11-03
Tue
1'114.05
```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang sesuai.

```
>day(vt[1]),  ...
>indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])),  ...
>strrepl(vt[3],"'","")()
```

```
7.3816e+05
2
1114
```

Menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari baris data. Asumsikan kita memiliki baris berikut dokumen HTML.

```
>line="<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>"
```

```
<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>
```

Untuk mengekstrak ini, kami menggunakan ekspresi reguler, yang mencari

```
- kurung tutup >,
- string apa pun yang tidak mengandung tanda kurung dengan
```

sub-pertandingan "(...)",

```
- braket pembuka dan penutup menggunakan solusi terpendek,
- lagi string apa pun yang tidak mengandung tanda kurung,
- dan kurung buka <.
```

Ekspresi reguler agak sulit dipelajari tetapi sangat kuat.

```
>{pos,s,vt}=strxfind(line,">([^<>]+)<.+?>([^<>]+)<");
```

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-pertandingan.

```
>for k=1:length(vt); vt[k](), end;
```

```
1145.5
5.6
```

Berikut adalah fungsi, yang membaca semua item numerik antara <td> dan </td>.

```
>function readtd (line) ...
```

```
  v=[]; cp=0;
  repeat
     {pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);
     until pos==0;
     if length(vt)>0 then v=v|vt[1]; endif;
     cp=pos+strlen(s);
  end;
  return v;
  endfunction
```

```
>readtd(line+"<td>non-numerical</td>")
```

```
  1145.45
  5.6
  -4.5
  non-numerical
```

Membaca dari Web

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.
Dalam contoh, kami membaca versi saat ini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." dalam sebuah judul.

```
>function readversion () ...
```

```
  urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html");
  repeat
    until urleof();
    s=urlgetline();
    k=strfind(s,"Version ",1);
    if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif;
  end;
  urlclose();
  endfunction
```

```
>readversion
```

```
  Version 2022-05-18
```

Input dan Output Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke file atau ke baris perintah.

```
>writevar(pi,"mypi");
```

```
  mypi = 3.141592653589793;
```

Untuk pengujian, kami membuat file Euler di direktori kerja EMT.

```
>file="test.e"; ...
>writevar(random(2,2),"M",file); ...
>printfile(file,3)
```

```
M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];
```

Kita sekarang dapat memuat file. Ini akan mendefinisikan matriks M.

```
>load(file); show M,
```

```
M =
  0.59918   0.79603
  0.51672   0.29967
```

Omong-omong, jika writevar() digunakan pada variabel, itu akan mencetak definisi variabel dengan nama variabel ini.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];
inch$ = 0.0254;
```

Kita juga bisa membuka file baru atau menambahkan file yang sudah ada. Dalam contoh kami menambahkan ke file yang dihasilkan sebelumnya.

```
>open(file,"a"); ...
>writevar(random(2,2),"M1"); ...
>writevar(random(3,1),"M2"); ...
>close();
>load(file); show M1; show M2;
```

```
M1 =
  0.30287   0.15372
   0.7504   0.75401
M2 =
  0.27213
 0.053211
  0.70249
```

Untuk menghapus file apa pun, gunakan fileremove().

```
>fileremove(file);
```

Vektor baris dalam file tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita buat file seperti itu, menulis setiap baris satu per satu dengan writeln().

```
>open(file,"w"); writeln("M = ["); ...
>for i=1 to 5; writeln(""+random()); end; ...
>writeln("];"); close(); ...
>printfile(file)
```

```
M = [
0.344851384551
0.0807510017715
0.876519562911
0.754157709472
0.688392638934
];
```

```
>load(file); M
```

```
[0.34485,  0.080751,  0.87652,  0.75416,  0.68839]
```

catatan : ketika mengenter perintah-perintah diatas ternyata hasil yang didapatkan berbeda-beda

Latihan soal

Nomor 1
Carilah rata-rata dan standar deviasi beserta plot dari data berikut
X = 1000,1500,1700,2500,3500,4000

```
>X=[1000,1500,1700,2500,3500,4000]; ...
>mean(X), dev(X),
```

```
2366.7
1186
```

```
>aspect(1.5); boxplot(X):
```

images/EMTStatistika_22305141008_Akhmad Rizki Khamid-039.png

Nomor 2
Misalkan diberikan data skor hasil statistika dari 14 orang mahasiswa sebagai berikut:
50,92,68,72,84,80,96,64,70,48,88,66,56,84
Tentukan rata-rata dari data tersebut!

```
>X=[62,65,58,90,75,79,82,91,75,75,75,95]
```

```
[62,  65,  58,  90,  75,  79,  82,  91,  75,  75,  75,  95]
```

```
>mean(X)
```

```
76.833
```