NAMA : NADZWA SRI AZIJZAH

NIM : 23030630043

KELAS : MATEMATIKA E 2023

TUGAS Materi EMT ALJABAR - STATISTIKA

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 Indonesia;
- Mencoba contoh-contoh peritungan (perintah EMT) dengan cara meng-ENTER setiap perintah EMT yang ada (pindahkan kursor ke baris perintah)
- Jika perlu Anda dapat memodifiksi 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 eksk 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 yang diakhiri dengan tanda titik koma ";" atau koma ",". Tanda titik koma akan menghentikan hasil yang akan ditampilkan. Sehingga koma setelah perintah terakhir bisa dihilangkan.

Baris perintah berikut hanya akan menampilkan hasil dari ekspresi, melainkan bukan penugasan atau pun peritah format.

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

```
  16.7551608191
```

Perintah harus dipisahkan dengan spasi. Baris perintah berikut akan mencetak dua hasilnya.

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

```
  50.2654824574
  100.530964915
```

Baris perintah dieksekusi sesuai uruan saat pengguna menekan tombol enter. Jadi, setiap kali Anda akan mendapatkan nilai baru.

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

```
  0.540302305868
```

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

```
  0.857553215846
```

Jika dua baris kode dihubungkan dengan "..." maka kedua baris tersebut 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
```

Tidak masalah menggunakan beberapa baris. Pastikan baris tersebut diakhiri dengan "...".

```
>x := 1.5; // komentar ditulis di sini sebelum ...
>repeat xnew:=(x+2/x)/2; until xnew~=x; ...
>  x := xnew; ...
>end; ...
>x,
```

```
  1.41421356237
```

Struktur kondisional juga berfungsi.

```
>if E^pi>pi^E; then "Good Job, Nadzwa!", endif;
```

```
  Good Job, Nadzwa!
```

Ini juga merupakan cara yang baik untuk membagi perintah panjang menjadi dua atau lebih baris. Anda bisa teksn Ctrl+Return untuk membagi baris menjadi dua di posisi kursor saat ii=ni, atau Ctrl+Back untuk menghubungkan baris-baris tersebut.

Untuk melipat semua baris multi-lines, tekanlah Ctrl+L. Kemudian baris-baris berikutnya hanya akan terlihat jika salah satunya mendapatkan fokus. Untuk melipat satu baris multi-line, mulailah dengan "%+ ".

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

Baris yang dimulai dengan %% akan sepenuhnya tidak terlihat.

81

Euler mendukung penggunaan loop dalam baris perintah, selama semuanya muat dalam satu baris atau beberapa baris. Dalam program, batasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut, lihat pengantar berikut.

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

```
  Cannot assign to a value of type string.
  Error in:
   %+ x=4+5; %+ x=4+5; %+ x=4+5; %+ x=4+5; %+ x=4+5; %+ x=4+5; %+ ...
                ^
```

Tidak masalah menggunakan beberapa baris. Pastikan baris tersebut 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 kondisional juga berfungsi.

```
>if E^pi>pi^E; then "Good Job, Nadzwa!", endif;
```

```
  Good Job, Nadzwa!
```

Ketika Anda menjalankan sebuah perintah, kursor bisa berada di posisi mana saja dalam baris perintah. Anda bisa kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau Anda bisa mengklik bagian komentar di atas perintah untuk menuju ke perintah tersebut.

Saat Anda memindahkan kursor sepanjang baris, pasangan tanda kurung atau tanda kurung bulat akan disorot. Selain itu, perhatikan baris status. Setelah tanda kurung pembuka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol enter.

```
>sqrt(sin(10°)/cos(20°))
```

```
  0.429875017772
```

Untuk melihat bantuan untuk perintah terbaru, buka jendela bantuan dengan menekan F1. Di sana, Anda bisa memasukkan teks untuk mencari. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda bisa menekan tombol escape untuk membersihkan baris tersebut atau menutup jendela bantuan.

Anda bisa mengklik dua kali pada perintah apa pun untuk membuka bantuan untuk perintah tersebut. Cobalah mengklik dua kali pada perintah exp di bawah ini di baris perintah.

```
>exp(log(5.5))
```

```
  5.5
```

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

Basic Syntax atau Sintaks Dasar

Euler mengenal fungsi matematika yang umum. Seperti yang telah Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilai, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt di Euler. Tentu saja, x^(1/2) juga bisa digunakan.

Untuk mengatur variabel, gunakan "=" atau ":=". Untuk kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak mempengaruhi. Namun, spasi antara perintah diharapkan ada.

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

```
>g:=10; t:=3; 1/2*g*t^2
```

```
  45
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

Anda harus menyetel tanda kurung yang benar dan menggunakan / untuk pecahan. Perhatikan tanda kurung yang disorot untuk bantuan. Perlu dicatat bahwa konstanta Euler e disebut E dalam EMT.

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

```
  8.77908249441
```

Untuk menghitung ekspresi yang rumit seperti

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

Anda perlu memasukkannya dalam bentuk baris.

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

```
  23.2671801626
```

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

Hasil dari perhitungan ini adalah angka floating point. Secara default, hasil tersebut dicetak dengan akurasi sekitar 12 digit. Pada baris perintah berikutnya, kita juga akan belajar bagaimana kita bisa merujuk ke hasil sebelumnya dalam baris yang sama.

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

```
  0.47619047619
  10/21
```

Sebuah perintah Euler bisa berupa ekspresi atau perintah primitif. Ekspresi terdiri dari operator dan fungsi. Jika perlu, harus mengandung tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, menambahkan tanda kurung adalah ide yang baik. Perlu dicatat bahwa EMT menunjukkan tanda kurung pembuka dan penutup saat mengedit baris perintah.

```
>(cos(pi/6)+1)^3*(sin(pi/2)+1)^2
```

```
  25.9903810568
```

Operator numerik di Euler meliputi

+ penjumlahan atau operator plus
- pengurangan atau operator minus
*, /
. produk matriks
a^b pangkat untuk a positif atau b bilangan bulat (a**b juga berfungsi) n! operator faktorial

dan masih banyak lagi.

Berikut beberapa fungsi yang mungkin Anda butuhkan. 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, misalnya ln untuk log.

```
>ln(E^2), arctan(tan(1))
```

```
  2
  1
```

```
>sin(90°)
```

```
  1
```

Pastikan untuk menggunakan tanda kurung (kurung bulat) setiap kali ada keraguan tentang urutan eksekusi! Yang berikut ini tidak sama dengan (2^3)^4, yang merupakan default untuk 2^3^4 dalam EMT (beberapa sistem numerik melakukannya dengan cara sebaliknya).

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

```
  65536
  256
  65536
```

Bilangan Rill

Tipe data utama di Euler adalah bilangan riil. Bilangan riil direpresentasikan dalam format IEEE dengan akurasi sekitar 16 digit desimal.

```
>longest 7/3
```

```
        2.333333333333333
```

Representasi internal ganda memerlukan 8 byte.

```
>printdual(7/3)
```

```
  1.0010101010101010101010101010101010101010101010101011*2^1
```

```
>printhex(7/3)
```

```
  2.5555555555556*16^0
```

String

Sebuah string di Euler didefinisikan dengan "...".

```
>"Nadzwa Sri Azijzah."
```

```
  Nadzwa Sri Azijzah.
```

String dapat digabungkan dengan | atau dengan +. Ini juga berlaku untuk angka, yang dikonversi menjadi string dalam kasus tersebut.

```
>"Luas suatu daerah dengan radius " + 3 + " adalah " + pi*2 + " cm^2."
```

```
  Luas suatu daerah dengan radius 3 adalah 6.28318530718 cm^2.
```

Fungsi print juga mengonversi angka menjadi string. Fungsi ini dapat menerima jumlah digit dan jumlah tempat (0 untuk output padat), serta idealnya sebuah unit.

```
>"Tinggi badanku : " + print((1+sqrt(100000))/2,6,0)
```

```
  Tinggi badanku : 158.613883
```

Ada string khusus bernama none, yang tidak dicetak. String ini dikembalikan oleh beberapa fungsi ketika hasilnya tidak penting. (Ini dikembalikan secara otomatis jika fungsi tersebut tidak memiliki pernyataan return.)

```
>none
```

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

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

```
  1234.5
```

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

```
>v:=["Nadzwa","suka","melukis"]
```

```
  Nadzwa
  suka
  melukis
```

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

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

```
  Nadzwa
  suka
  melukis
  Nadzwa
  suka
  melukis
```

String dapat berisi karakter Unicode. Secara internal, string ini menggunakan 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

Dalam komentar, entitas yang sama seperti a, ß, dll. dapat digunakan. Ini bisa menjadi alternatif cepat untuk LaTeX. (Detail lebih lanjut tentang 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 sebuah variabel menjadi string Unicode.

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

```
  We have =.
```

Juga memungkinkan untuk menggunakan entitas numerik.

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

```
  Ähnliches
```

Nilai Boolean

Nilai Boolean direpresentasikan dengan 1=true atau 0=false di Euler. % String dapat dibandingkan, sama seperti angka.

```
>5<9, "jeruk"<"mangga"
```

```
  1
  1
```

"and" adalah operator "&&" dan "or" adalah operator "||", seperti

dalam bahasa C. (Kata "and" dan "or" hanya dapat digunakan dalam kondisi untuk "if".)

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

```
  1
```

Operator Boolean mengikuti aturan bahasa matriks.

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

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

Anda bisa menggunakan fungsi nonzeros() untuk mengekstrak elemen tertentu dari sebuah 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 Output

Format output/hasil default EMT mencetak 12 digit. Untuk memastikan bahwa kita melihat format default, kita mereset format tersebut.

```
>defformat; pi/2
```

```
  1.57079632679
```

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

```
>longest pi
```

```
        3.141592653589793
```

Berikut adalah representasi internal heksadesimal dari angka ganda.

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
      0.33333
      3.14159
      0.84147
```

Default-nya adalah format(12).

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

```
  0.333333333333
```

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

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

```
  3.1416
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
  3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran yang paling penting.
shortestformat shortformat longformat, longestformat
format(length,digits) goodformat(length)
fracformat(length)
defformat

Akurasi internal EMT sekitar 16 tempat desimal, sesuai dengan standar IEEE. Angka 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
```

Default-nya adalah defformat().

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

Ada operator singkat yang hanya mencetak satu nilai. Operator "longest" akan mencetak semua digit valid dari sebuah angka.

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

```
        4.934802200544679
```

Ada juga operator singkat untuk mencetak hasil dalam format pecahan. Kita 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. Kesalahan ini bertambah sedikit demi sedikit, seperti yang Anda lihat dalam perhitungan berikut.

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

```
   -1.110223024625157e-16
```

Namun dengan "longformat" 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
```

Expressions

Nama atau teks (string) bisa digunakan untuk menyimpan persamaan matematika yang bisa dihitung oleh EMT. Untuk melakukannya, gunakan tanda kurung setelah persamaannya. Jika ingin menggunakan teks sebagai persamaan, gunakan aturan penamaan seperti "fx" atau "fxy", dan sebagainya. Persamaan ini memiliki prioritas lebih tinggi dibandingkan fungsi.

Variabel global juga bisa digunakan saat melakukan perhitungan.

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

```
        12.56637061435917
```

Parameter diberikan ke x, y, dan z secara berurutan. Parameter tambahan bisa ditambahkan menggunakan parameter yang sudah ditetapkan.

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

```
  -0.919535764538
```

Perlu diingat bahwa ekspresi akan selalu menggunakan variabel global, meskipun ada variabel dengan nama yang sama di dalam fungsi. (Jika tidak, perhitungan ekspresi dalam fungsi bisa menghasilkan hasil yang 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 ingin menggunakan nilai lain untuk "at" selain dari nilai global, kamu perlu menambahkan "at=nilai".

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

```
  45
```

Sebagai referensi, perlu dicatat bahwa kumpulan panggilan (dibahas di tempat lain) bisa berisi ekspresi. Jadi, kita bisa membuat contoh di atas seperti berikut.

am fungsi bisa menghasilkan hasil yang membingungkan bagi pengguna yang memanggil fungsi tersebut.)

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

```
  45
```

Ekspresi pada x sering digunakan seperti fungsi.
erlu diingat bahwa jika kamu mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global, variabel tersebut akan dihapus untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

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

```
  12
```

Sebagai aturan, ekspresi simbolik atau numerik sebaiknya diberi nama seperti fx, fxy, dan sebagainya. Skema penamaan ini tidak boleh digunakan untuk fungsi.

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

Ada bentuk khusus dari ekspresi yang memungkinkan penggunaan variabel apa pun sebagai parameter tanpa nama saat menghitung ekspresi, tidak hanya "x", "y", dan sebagainya. Untuk ini, mulailah ekspresi dengan "@(variabel) ...".

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

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

Ini memungkinkan untuk memanipulasi ekspresi dengan variabel lain dalam fungsi EMT yang memerlukan ekspresi di "x".

Cara paling sederhana untuk mendefinisikan fungsi adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utamanya adalah x, ekspresi tersebut bisa dihitung seperti fungsi.

Seperti yang terlihat pada contoh berikut, variabel global dapat terlihat selama perhitungan.

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

```
  -0.475
```

Semua variabel lain dalam ekspresi bisa ditentukan dalam perhitungan menggunakan parameter yang telah ditetapkan.

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

```
  -0.425
```

Ekspresi tidak harus simbolik. Ini diperlukan jika ekspresi mengandung fungsi yang hanya dikenal di kernel numerik, bukan di Maxima.

Matematika Simbolik

EMT melakukan matematika simbolik dengan bantuan Maxima. Untuk detail lebih lanjut, mulailah dengan tutorial berikut atau lihat referensi untuk Maxima. Para ahli Maxima harus memperhatikan bahwa ada perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik di EMT.

Matematika simbolik terintegrasi secara mulus ke dalam Euler dengan menggunakan &. Setiap ekspresi yang diawali dengan & adalah ekspresi simbolik. Ekspresi tersebut dihitung dan dicetak oleh Maxima.

Pertama-tama, Maxima memiliki aritmetika "tak hingga" yang dapat menangani angka yang sangat besar.

```
>$&33!
```

Dengan cara ini, Anda bisa menghitung hasil besar secara tepat. Mari kita hitung

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

```
>$& 33!/(24!*10!) // nilai C(33,10)
```

Tentu saja, Maxima memiliki fungsi yang lebih efisien untuk ini (begitu juga dengan bagian numerik dari EMT).

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

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali pada fungsi tersebut. Misalnya, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima yang disediakan oleh pembuat program tersebut.

Kamu juga akan belajar bahwa hal berikut ini juga bisa dilakukan.

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

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

Jika kamu ingin mengganti x dengan nilai tertentu, gunakan "with".

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

Dengan cara ini, kamu bisa menggunakan solusi dari satu persamaan dalam persamaan lainnya.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Ini disebabkan oleh sebuah tanda simbolik khusus dalam string.

Seperti yang akan kamu lihat dalam contoh sebelumnya dan berikutnya, jika kamu memiliki LaTeX terinstal, kamu bisa mencetak ekspresi simbolik dengan LaTeX. Jika tidak, perintah berikut akan menghasilkan pesan kesalahan.

Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan $ di depan & (atau kamu bisa menghilangkan &) sebelum perintah. Jangan jalankan perintah Maxima dengan $, jika kamu tidak memiliki LaTeX terinstal.

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

Ekspresi simbolik diproses oleh Euler. Jika kamu memerlukan sintaks yang kompleks dalam satu ekspresi, kamu bisa membungkus ekspresi tersebut dalam "...". Menggunakan lebih dari sekadar ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.

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

25

Untuk melengkapi, perlu dicatat bahwa ekspresi simbolik bisa digunakan dalam program, tetapi harus dibungkus dalam tanda kutip. Selain itu, lebih efektif untuk memanggil Maxima saat waktu kompilasi jika memungkinkan.

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

Sekali lagi, % mengacu pada hasil sebelumnya.

Untuk mempermudah, kita simpan solusi ke dalam variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

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

Ekspresi simbolik bisa digunakan dalam ekspresi simbolik lainnya.

```
>$&factor(diff(fx,x))
>&factor(15!)
```

$$1307674368000$$

```
>::: factor(30!)
```

$$2^{26}\ 3^{14}\ 5^{7}\ 7^{4}\ 11^{2}\ 13^{2}\ 17\ 19\ 23\ 29$$

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

$$2^{18}\ 3^{8}\ 5^{4}\ 7^{2}\ 11\ 13\ 17\ 19$$

Jika kamu ahli dalam Maxima, kamu mungkin ingin menggunakan sintaks asli Maxima. Kamu bisa melakukannya dengan ":::"

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

$$g^{2}$$

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

$$x^{3}\ E^{x}$$

Variabel seperti itu bisa digunakan dalam ekspresi simbolik lainnya. Perhatikan bahwa dalam perintah berikut, sisi kanan dari &= dihitung terlebih dahulu sebelum penugasan ke Fx.

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

$$125\ E^5$$

$$18551.64488782208$$

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

Untuk mendapatkan kode LaTeX dari sebuah ekspresi, kamu bisa menggunakan perintah tex.

```
>tex(fx)
```

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

Ekspresi simbolik bisa dihitung seperti ekspresi numerik.

```
>fx(0.5)
```

```
  0.206090158838
```

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

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

Penugasan juga bisa bersifat simbolik.

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

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

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

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

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

```
  1.56155281281
```

Nilai numerik dari solusi simbolik bisa dihitung dengan mengevaluasi hasil simbolik tersebut. Euler akan mengabaikan penugasan seperti x= dan sebagainya. Jika kamu tidak memerlukan hasil numerik untuk perhitungan selanjutnya, kamu juga bisa 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, kamu bisa menggunakan "with" dan sebuah indeks.

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

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

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

Ekspresi simbolik bisa memiliki tanda (flags), yang menunjukkan perlakuan khusus di Maxima. Beberapa tanda bisa digunakan sebagai perintah juga, sementara yang lain tidak bisa. Tanda ditambahkan dengan "|" (bentuk yang lebih baik 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

Di EMT, fungsi adalah program yang didefinisikan dengan perintah "function". Fungsi bisa berupa satu baris atau beberapa baris.
Fungsi satu baris bisa bersifat numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan ":=".

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

Untuk gambaran umum, kami menunjukkan semua kemungkinan definisi untuk fungsi satu baris. Fungsi ini bisa dihitung seperti fungsi bawaan Euler lainnya.

```
>f(4)
```

```
  16.4924225025
```

Fungsi ini juga akan berfungsi untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut telah diproses untuk vektor.

```
>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 bisa digambar. Alih-alih memberikan ekspresi, kita hanya perlu memberikan nama fungsi.

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

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

```
  0.786151377757
```

Secara default, jika kamu perlu menimpa fungsi bawaan, kamu harus menambahkan kata kunci "overwrite". Menimpa fungsi bawaan bisa berbahaya dan dapat menyebabkan masalah bagi fungsi lain yang bergantung padanya.

Kamu masih bisa memanggil fungsi bawaan dengan menggunakan "_..." jika itu adalah fungsi di inti Euler.

```
>function overwrite sin (x) := _sin(x°) // redine sine in degrees
>sin(90)
```

```
  1
```

Lebih baik kita hapus penetapan ulang fungsi sin ini.

```
>forget sin; sin(pi/2)
```

```
  1
```

Parameter Bawaan

Fungsi numerik bisa memiliki parameter bawaan.

```
>function f(x,a=4) := a*x^2
```

Jika parameter ini diabaikan, nilai bawaan akan digunakan.

```
>f(6)
```

```
  144
```

Menetapkannya akan menggantikan nilai bawaan.

```
>f(6,7)
```

```
  252
```

Parameter yang ditetapkan juga menggantikan nilai bawaan. Ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

```
>f(6,a=4)
```

```
  144
```

Jika sebuah variabel bukan parameter, variabel tersebut harus bersifat global. Fungsi satu baris bisa mengakses variabel global.

```
>function f(x) := a*x^2
>a=4; f(2)
```

```
  16
```

Namun, parameter yang ditetapkan akan menggantikan nilai global.
Jika argumen tidak ada dalam daftar parameter yang sudah ditentukan, ia harus dideklarasikan dengan ":="!

```
>f(3,a:=6)
```

```
  54
```

Fungsi simbolik didefinisikan dengan "&=". Mereka didefinisikan di Euler dan Maxima, dan berfungsi di kedua lingkungan tersebut. Ekspresi yang mendefinisikannya diproses melalui Maxima sebelum definisi diterapkan.

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

Fungsi simbolik bisa digunakan dalam ekspresi simbolik.

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

Fungsi simbolik juga bisa digunakan dalam ekspresi numerik. Tentu saja, ini hanya akan berhasil jika EMT dapat menginterpretasikan semua yang ada di dalam fungsi tersebut.

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

```
  178.635099908
```

Fungsi simbolik bisa digunakan untuk mendefinisikan fungsi simbolik atau ekspresi lainnya.

```
>function G(x) &= factor(integrate(g(x),x)); $&G(c) // integrate: mengintegralkan
>solve(&g(x),0.5)
```

```
  0.703467422498
```

Yang berikut ini juga berhasil, karena Euler menggunakan ekspresi simbolik dalam fungsi g jika tidak menemukan variabel simbolik g, dan jika ada 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)
```

```
  625
```

```
>$&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 &=, fungsi tersebut bersifat simbolik dan bisa digunakan dalam ekspresi simbolik lainnya.

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

Dengan :=, fungsi tersebut bersifat numerik. Contoh yang baik adalah integral tertentu seperti

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

yang tidak bisa dihitung secara simbolik.

Jika kita mendefinisikan ulang fungsi dengan kata kunci "map," fungsi tersebut dapat digunakan untuk vektor x. Secara internal, fungsi akan 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 bisa memiliki nilai bawaan untuk parameter.

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

Sekarang fungsi tersebut bisa dipanggil dengan atau tanpa parameter "base".

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

```
  2
  6.7
```

Selain itu, kamu bisa menggunakan parameter yang ditetapkan.

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

```
  2
```

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

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

Fungsi simbolik seperti itu bisa digunakan untuk variabel simbolik.
Tapi fungsi tersebut juga bisa digunakan untuk vektor numerik.

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

```
  17
```

Ada juga fungsi yang sepenuhnya simbolik, yang tidak bisa digunakan secara numerik.

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

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

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

Namun, tentu saja, fungsi ini bisa 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 merangkum:
&= mendefinisikan fungsi simbolik,
= mendefinisikan fungsi numerik,
&= mendefinisikan fungsi yang sepenuhnya simbolik.

Menyelesaikan Ekspresi

Ekspresi bisa diselesaikan secara numerik dan simbolik.

Untuk menyelesaikan ekspresi sederhana dengan satu variabel, kita bisa menggunakan fungsi solve(). Fungsi ini memerlukan nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode sekant.

```
>solve("x^2-2",3)
```

```
  1.41421356237
```

Ini juga berlaku untuk ekspresi simbolik. Ambil contoh 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 di mana polinomialnya adalah 2. Dalam solve(), nilai target default y=0 bisa diubah dengan variabel yang ditetapkan. Kita gunakan y=2 dan periksa dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
  0.966715594851
  2
```

Menyelesaikan ekspresi simbolik dalam bentuk simbolik menghasilkan daftar solusi. Kita menggunakan pemecah simbolik 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 ekspresi.

```
>longest sol()
```

```
      -0.6180339887498949       1.618033988749895
```

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

```
>
```

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

Menyelesaikan sistem persamaan secara simbolik bisa dilakukan dengan vektor persamaan dan pemecah simbolik solve(). Jawabannya adalah daftar dari daftar persamaan.

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

Fungsi f() bisa mengakses variabel global. Namun, seringkali kita ingin menggunakan parameter lokal.

```
  a^x-x^a = 0.1
```

with a=3.

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

Salah satu cara untuk menyertakan parameter tambahan ke f() adalah dengan menggunakan daftar yang berisi nama fungsi dan parameter (cara lainnya adalah dengan menggunakan parameter titik koma).

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

```
  2.54116291558
```

Ini juga berlaku untuk ekspresi. Namun, dalam hal ini, elemen daftar yang bernama harus digunakan. (Lebih lanjut tentang daftar bisa ditemukan di tutorial tentang sintaks EMT).

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

```
  2.54116291558
```

Menyelsaikan Pertidaksamaan

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

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

```
>A=[2,3;4,5]
```

```
            2            3
            4            5
```

Produk matriks dinyatakan dengan titik.

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

```
            3
            4
```

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

```
  [3,  4]
```

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

```
      -2.5         1.5
         2          -1
```

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

```
        18
        32
```

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

```
         1           0
         0           1
```

Poin utama dari bahasa matriks adalah bahwa semua fungsi dan operator bekerja pada setiap elemen secara terpisah.

```
>A.A
```

```
        16          21
        28          37
```

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

```
         4           9
        16          25
```

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

```
       116         153
       204         269
```

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

```
       116         153
       204         269
```

```
>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)
```

```
  0.666667           1
         1        1.25
```

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

```
           -1.5
              2
```

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

```
           -1.5
              2
```

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

```
              1             0
              0             1
```

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

```
              1             0
              0             1
```

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

```
              4             9
             16            25
```

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

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

```
              9
             16
```

Jika salah satu operand adalah vektor atau skalar, itu akan diperluas dengan cara yang alami.

```
>2*A
```

```
              4             6
              8            10
```

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

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

```
              2             6
              4            10
```

Jika operandnya adalah vektor baris, elemennya diterapkan ke semua kolom A.

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

```
                4                9
                8               15
```

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

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

```
                1                2
                1                2
```

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

```
                2                6
                4               10
```

Ini juga berlaku untuk dua vektor di mana satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kita menghitung i*j untuk i dan j dari 1 hingga 5. Triknya adalah 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!

```
>(1:6).(1:6)' // hasilkali vektor baris dan vektor kolom
```

```
  91
```

```
>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 bisa 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 yang berisi 0 dan 1, di mana 1 menunjukkan 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 seperti itu, "nonzeros" memilih elemen-elemen yang tidak nol.
Dalam hal ini, kita mendapatkan indeks dari semua elemen yang lebih besar dari 50.

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

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

Tentu saja, kita bisa 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 temukan semua kuadrat dari angka 1 hingga 1000 yang merupakan 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. Secara internal, ia menggunakan floating point presisi ganda. Namun, sering kali sangat berguna.
Kita bisa memeriksa keprimaan.
Mari kita cari tahu berapa banyak kuadrat ditambah 1 yang merupakan bilangan prima.

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

```
  112
```

Fungsi nonzeros() hanya bekerja 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
```

Fungsi ini mengembalikan indeks elemen-elemen yang tidak nol.

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

```
  endfunction
```

Indeks-indeks ini dapat digunakan untuk mengatur elemen-elemen ke nilai tertentu.

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

```
  Variable or function k not found.
  Error in:
  mset(A,k,0) //mengganti elemen2 suatu matriks pada indeks tert ...
          ^
```

Fungsi mset() juga bisa mengatur elemen-elemen pada indeks-indeks tersebut dengan entri dari matriks lain.

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

```
  Variable or function k not found.
  Error in:
  mset(A,k,-random(size(A))) ...
          ^
```

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

```
  Variable or function k not found.
  Error in:
  mget(A,k) ...
          ^
```

Dan memungkinkan untuk mendapatkan elemen-elemen tersebut dalam sebuah vektor.

Fungsi berguna lainnya adalah extrema, yang mengembalikan nilai minimal dan maksimal di setiap baris matriks beserta posisinya.

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

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

```
>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 bisa mengekstrak indeks-indeks tersebut dan menggunakan informasi ini untuk mengambil elemen dari 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 (Membangun Matriks)

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

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

Demikian pula, kita bisa menempelkan matriks di samping matriks lain, jika keduanya memiliki jumlah baris yang sama.

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

```
  Commands must be separated by semicolon or comma!
  Found: v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:
  You can disable this in the Options menu.
  Error in:
  v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3; v_v v=1:3;  ...
            ^
```

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

Ada pengecualian untuk aturan ini. Sebuah angka riil yang ditempelkan pada matriks akan digunakan sebagai kolom yang diisi dengan angka riil tersebut.

```
>A|1
```

```
      0.765761      0.401188      0.406347      0.267829             1
       0.13673      0.390567      0.495975      0.952814             1
      0.548138      0.006085      0.444255      0.539246             1
```

Dimungkinkan untuk membuat matriks dari vektor baris dan vektor kolom.

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

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

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

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

Tujuan utamanya adalah untuk menginterpretasikan vektor ekspresi sebagai vektor kolom.

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

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

Untuk mendapatkan ukuran matriks A, kita bisa menggunakan fungsi-fungsi berikut.

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

```
  2
  4
  [2,  4]
  4
```

Untuk vektor, ada fungsi length().

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

```
  9
```

Ada banyak fungsi lain yang menghasilkan matriks.

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

```
                1               1
                1               1
```

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

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

```
       0.896177       0.251157
       0.153047       0.126917
```

Berikut adalah fungsi berguna lainnya yang mengubah struktur elemen-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 bisa menggunakan fungsi ini dan fungsi dup untuk menulis fungsi rep(), yang mengulang vektor sebanyak kali.

```
>function rep(v,n) := redim(dup(v,n),1,n*cols(v))
>rep(1:3,5)
```

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

Fungsi multdup() menggandakan elemen-elemen dari sebuah 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() membalik urutan baris atau kolom dari sebuah matriks. Misalnya, fungsi flipx() membalik secara horizontal.

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

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

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

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

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

Fungsi khusus adalah drop(v, i), yang menghapus elemen-elemen dengan indeks dalam i dari vektor v.

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

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

Perhatikan bahwa vektor dalam drop(v, i) merujuk pada indeks elemen di i bukan nilai elemen tersebut. Jika Anda ingin menghapus elemen, Anda perlu menemukan elemen-elemen tersebut terlebih dahulu. Fungsi indexof(v, x) dapat digunakan untuk menemukan elemen x dalam vektor terurut v.

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

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

Seperti yang Anda lihat, tidak ada masalah jika menyertakan indeks yang berada di luar jangkauan (seperti 0), indeks ganda, atau indeks yang tidak terurut.

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

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

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

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

```
                1               0               0               0               0
                0               1               0               0               0
                0               0               1               0               0
                0               0               0               1               0
                0               0               0               0               1
```

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

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

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

Perhatikan bahwa kita tidak mengubah matriks A. Kita mendapatkan matriks baru sebagai hasil dari setdiag().
Berikut adalah fungsi yang mengembalikan matriks tri-diagonal.

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

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

Sekarang kita bisa mengekstrak diagonalnya.

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

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

Misalnya, kita bisa membagi matriks dengan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan ke matriks baris demi baris.

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

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

Faktoriasi

Hampir semua fungsi di Euler juga bekerja untuk input matriks dan vektor, selama 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 bisa dengan mudah membuat tabel nilai. Ini adalah salah satu cara untuk memplot sebuah fungsi (alternatifnya menggunakan ekspresi).

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

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

EMT memperluas operator untuk skalar, vektor, dan matriks dengan cara yang jelas.
Misalnya, vektor kolom dikalikan dengan vektor baris akan diperluas menjadi matriks, jika operator diterapkan. Dalam hal ini,
v adalah vektor yang ditranspos (vektor kolom).

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

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

Perhatikan bahwa ini berbeda dari produk matriks. Produk matriks dilambangkan dengan titik "." dalam EMT.

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

```
  55
```

Secara default, vektor baris dicetak dalam format kompak.

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

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

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

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

```
  5
  25
```

Untuk mentranspos matriks, kita menggunakan apostrof (').

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

```
             1
             2
             3
             4
```

Jadi, kita bisa menghitung matriks A dikalikan dengan 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 kuadrat dari vektor baris v. Hasilnya adalah matriks 1x1, yang berfungsi seperti angka riil.

```
>v.v'
```

```
  30
```

Ada juga fungsi norm (bersama dengan banyak fungsi lain dalam Aljabar Linier).

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

```
  30
```

Operator dan fungsi mematuhi bahasa matriks Euler. Berikut ringkasan aturan yang berlaku:

- Fungsi yang diterapkan pada vektor atau matriks akan diterapkan pada setiap elemen secara individual.

- Operator yang beroperasi pada dua matriks dengan ukuran yang sama akan diterapkan secara berpasangan pada elemen-elemen matriks tersebut.

- Jika kedua matriks memiliki dimensi yang berbeda, keduanya akan diperluas secara logis sehingga ukurannya sama.

Contohnya, nilai skalar yang dikalikan dengan vektor akan mengalikan nilai tersebut dengan setiap elemen vektor. Demikian juga, matriks yang dikalikan dengan vektor (menggunakan *, bukan .) akan memperluas vektor ke ukuran matriks dengan menggandakannya.

Berikut adalah contoh sederhana dengan operator ^.

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

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

Berikut ini adalah contoh yang lebih rumit. Vektor baris yang dikalikan dengan vektor kolom akan memperluas keduanya dengan menggandakannya. Ini berarti elemen-elemen dari vektor baris dan kolom akan dipasangkan secara otomatis untuk melakukan operasi penggandaan antara elemen-elemen tersebut.

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

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

Perlu dicatat bahwa produk skalar menggunakan produk matriks, bukan operator *. Dalam operasi produk skalar, vektor baris dikalikan dengan vektor kolom melalui aturan perkalian matriks, di mana hasilnya adalah sebuah skalar (atau elemen tunggal) yang merupakan penjumlahan dari hasil perkalian elemen-elemen yang sesuai.

```
>v.v'
```

```
  14
```

Ada banyak fungsi untuk matriks. Berikut adalah daftar singkat. Konsultasikan dokumentasi untuk informasi lebih lanjut mengenai perintah-perintah ini:

```
sum,prod computes the sum and products of the rows
cumsum,cumprod does the same cumulatively
computes the extremal values of each row
extrema returns a vector with the extremal information
diag(A,i) returns the i-th diagonal
setdiag(A,i,v) sets the i-th diagonal
id(n) the identity matrix
det(A) the determinant
charpoly(A) the characteristic polynomial
eigenvalues(A) the eigenvalues
```

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

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

Operator : menghasilkan vektor baris dengan jarak yang sama, dengan opsi penentuan langkah (step size).

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

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

Untuk menggabungkan matriks dan vektor, tersedia operator "|" untuk penggabungan horizontal dan "_" untuk penggabungan vertikal.

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

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

Elemen-elemen dari sebuah 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 dari matriks.

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

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

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

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

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

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

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

```
>A{4}
```

```
  4
```

```
>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]
```

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

```
  Wrong argument.
  Cannot use a string matrix here.

  Try "trace errors" to inspect local variables after errors.
  deg:
      return x/pi*180;
  Error in:
  M = deg(w)|w|cos(w)|sin(w) ...
            ^
```

Menggunakan bahasa matriks, kita bisa menghasilkan beberapa tabel dari beberapa fungsi sekaligus.

Dalam contoh berikut, kita menghitung [j] untuk i dari 1 hingga n. Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel dari
t^i untuk satu. Artinya, matriks tersebut memiliki elemen-elemen

$$a_{i,j} = t_j^i, \quad 1 \le j \le 101, \quad 1 \le i \le n$$

Fungsi yang tidak dapat digunakan untuk input vektor perlu "divectorisasi". Ini dapat dicapai dengan kata kunci "map" dalam definisi fungsi. Dengan cara ini, fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.

Integrasi numerik integrate() hanya bekerja untuk batas interval skalar. Jadi, kita perlu menderivasikannya.

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

Kata kunci "map" membuat fungsi menjadi vektorisasi. Fungsi tersebut sekarang akan berfungsi untuk vektor angka.

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

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

Sub Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi kurung siku.

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

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

```
>A[2]
```

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

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

```
  2
```

Kita bahkan bisa mengurutkan ulang A menggunakan vektor indeks. Untuk lebih jelasnya, kita tidak mengubah A di sini, tetapi menghitung versi
A yang sudah diurutkan ulang.

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

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

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

```
                3
                6
                9
```

Sebagai alternatif, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir dari A.

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

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

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

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

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

```
                0              0             -1
                0              0              6
                7              8              9
```

Peringatan: Indeks yang melebihi batas akan mengembalikan matriks kosong atau pesan kesalahan, tergantung pada pengaturan sistem. Secara default, akan muncul pesan kesalahan. Namun, indeks negatif dapat digunakan untuk mengakses elemen matriks dari akhir.

```
>A[4]
```

```
  Row index 4 out of bounds!
  Error in:
  A[4] ...
      ^
```

Pengurutan dan Pengacakan

Fungsi sort() mengurutkan vektor baris.

Seringkali kita perlu mengetahui indeks dari vektor yang sudah diurutkan dalam vektor aslinya. Ini bisa digunakan untuk mengurutkan vektor lain dengan cara yang sama.

Mari kita acak urutan vektor.

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

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

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

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

Ini juga berlaku untuk vektor yang berisi 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
```

```
>ind
```

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

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

```
[9,  1,  7,  7,  8,  7,  9,  10,  6,  2]
Closing bracket missing in function call!
Error in:
intrandom(1,10,10), unique(% ...
                           ^
```

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar Linear

EMT memiliki banyak fungsi untuk menyelesaikan sistem linier, sistem jarang (sparse), atau masalah regresi. Untuk sistem linier Ax=b, Anda dapat menggunakan algoritma Gauss, invers matriks, atau pendekatan regresi linier. Operator A/b menggunakan versi dari algoritma Gauss.

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

```
      -4
     4.5
```

Untuk contoh lain, kita menghasilkan matriks berukuran 200x200 dan jumlah dari baris-barisnya. Kemudian kita menyelesaikan Ax=b menggunakan invers matriks. Kita mengukur kesalahan sebagai deviasi maksimum dari semua elemen dari 1, yang tentu saja adalah solusi yang benar.

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

```
2.848832281188152e-13
```

Jika sistem tidak memiliki solusi, penyesuaian 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
```

```
>det(A)
```

```
  0
```

Matriks Simbolik

Maxima memiliki matriks simbolik. Tentu saja, Maxima bisa digunakan untuk masalah aljabar linier sederhana seperti ini. Kita bisa mendefinisikan matriks untuk Euler dan Maxima dengan &:=, dan kemudian menggunakannya dalam ekspresi simbolik. Bentuk [...] yang biasa digunakan untuk mendefinisikan matriks juga bisa 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]; $A
>$&eigenvectors([a,1;1,a]), &%[2][1][1]
```

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

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

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

```
                1               4
                5               2
```

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

```
>&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.

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

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

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

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

```
                        1                        0
                        0                        1
```

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

```
  [16.1168,  -1.11684,  0]
```

```
>$&eigenvalues(@A)
```

Nilai Numerik dalam Ekspresi simbolik

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

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

```
              1        3.14159
              4              5
```

Masih terdapat perbedaan antara numerik dan simbolik membentuk. Saat menerjemahkan matriks ke bentuk simbolik, pecahan perkiraan real akan digunakan.

```
>mxmset(A); $&A
>$&bfloat(sqrt(2)), $&float(sqrt(2))
>&fpprec:=100; &bfloat(pi)
```

```
          3.14159265358979323846264338327950288419716939937510582097494\
  459230781640628620899862803482534211706 8b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolik apa pun yang 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 memecahkan masalah kehidupan nyata. Asumsikan Anda memiliki modal awal sebesar 5.000 (katakanlah dalam dolar).

```
>K=5000
```

```
  5000
```

```
>K*1.03
```

```
  5150
```

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

```
  150
```

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

```
  1.03
  5150
```

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

```
  6719.58189672
```

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

```
      6719.58
```

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

```
  Real 1 x 11 matrix

      5000.00     5150.00     5304.50     5463.64     ...
```

```
>short 0:10
```

```
  [0,  1,  2,  3,  4,  5,  6,  7,  8,  9,  10]
```

```
>VK=K*q^(0:10);
>function oneyear (K) := round(K*q,2)
> ...
>longest oneyear(1234.57), longest 1234.57*q
```

```
                1271.61
               1271.6071
```

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

```
  Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64      ...
```

```
>VKr'
```

```
     5000.00
     5150.00
     5304.50
     5463.64
     5627.55
     5796.38
     5970.27
     6149.38
     6333.86
     6523.88
     6719.60
```

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

```
     5150.00
     5000.00     5150.00     5304.50
```

```
>VKr[-1], VK[-1]
```

```
     6719.60
     6719.58
```

Menyelesaikan Persamaan

Sekarang kita mengambil fungsi yang lebih maju, yang masing-masing menambahkan tingkat uang tertentu tahun.

```
>function onepay (K) := K*q+R
> ...
>function onepay (K) := K*q+R
>R=200; iterate("onepay",5000,10)
```

```
  Real 1 x 11 matrix

     5000.00     5350.00     5710.50     6081.82      ...
```

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

```
  Real 1 x 11 matrix
      5000.00      4950.00      4898.50      4845.45      ...
```

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

```
  Real 1 x 51 matrix
      5000.00      4950.00      4898.50      4845.45      ...
```

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

```
         48.00
```

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

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

Fungsi iterate() mempunyai satu trik lagi. Ini dapat mengambil kondisi akhir sebagai sebuah 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 nilainya adalah 0 setelah 50 tahun. Berapa tingkat bunganya?

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

Langkah pertama adalah mendefinisikan fungsi yang melakukan iterasi sebanyak n kali. Kami tambahkan semua parameter ke fungsi ini.

```
>function f(K,R,P,n) := iterate("x*(1+P/100)+R",K,n;P,R)[-1]
```

Iterasinya sama seperti di atas

x_{n+1} = x_n \cdot \left(1+ \frac{P}{100}\right) + R

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.

Selain itu, kami hanya tertarik pada nilai terakhir. Jadi kita ambil indeksnya [-1].

Mari kita coba tes.

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

```
       -19.83
```

```
>solve("f(5000,-200,x,50)",3)
```

```
         3.15
```

per tahun. Kami mengambil nilai awal 3% untuk algoritma. Pemecahannya()

Kita bisa menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang bisa kita hapus per tahun sehingga modal awal habis setelah asumsi 20 tahun tingkat bunga 3% per tahun.

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

```
      -336.08
```

Solusi Simbolis Masalah Suku Bunga

Kita dapat menggunakan bagian simbolik Euler untuk mempelajari masalahnya. Pertama kita definisikan fungsi kami onepay() secara simbolis.

```
>function op(K) &= K*q+R; $&op(K)
>$&op(op(op(op(K)))), $&expand(%)
>&sum(q^k,k,0,n-1); $& % = ev(%,simpsum)
>function fs(K,R,P,n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K,R,P,n)
>longest f(5000,-200,3,47), longest fs(5000,-200,3,47)
```

```
       -19.82504734650985
       -19.82504734652684
```

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

```
        20.51
```

```
>$&limit(R(5000,0,x,10),x,0)
>fn &= solve(equ,n) | ratsimp; $&fn
```

Plot2D

* Menggambar Grafik 2D dengan EMT

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).

Plots Dasar

Ada beberapa fungsi dasar dalam membuat grafik. Ada koordinat layar, yang selalu berkisar antara 0 hingga 1024 di setiap sumbu, tidak peduli apakah layar tersebut berbentuk kotak atau tidak. Ada juga koordinat grafik, yang bisa diatur dengan fungsi setplot(). Cara pemetaan antara koordinat layar dan koordinat grafik bergantung pada jendela grafik yang sedang aktif. Misalnya, fungsi shrinkwindow() secara default menyisakan ruang untuk label sumbu dan judul grafik.

Dalam contoh ini, kita hanya menggambar beberapa garis acak dengan berbagai warna. Untuk detail lebih lanjut tentang fungsi-fungsi ini, pelajari fungsi-fungsi inti dari 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
>reset;
```

Penting untuk menahan grafik, karena perintah plot() akan menghapus jendela grafik yang ada.

Untuk menghapus semua yang telah kita buat, kita menggunakan perintah 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 lainnya, kita menggambar sebuah grafik kecil di dalam grafik yang lebih besar. Ini dilakukan dengan mendefinisikan jendela grafik yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela grafik. Kita perlu menambahkan margin jika diperlukan. Juga, perhatikan bahwa kita menyimpan dan mengembalikan jendela penuh, serta mempertahankan grafik yang sedang ada saat kita menggambar grafik kecil tersebut.

```
>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);
```

Untuk membuat plot dengan beberapa gambar, kita bisa menggunakan fungsi figure().

Aspek Plot

Secara default, plot menggunakan jendela plot berbentuk kotak. Kamu bisa mengubah ini menggunakan fungsi aspect(). Jangan lupa untuk mengatur ulang aspek setelahnya. Kamu juga bisa mengubah pengaturan default ini melalui menu dengan memilih "Set Aspect" untuk rasio aspek tertentu atau sesuai ukuran jendela grafis saat ini.

Namun, kamu juga bisa mengubahnya hanya untuk satu plot. Caranya adalah dengan mengubah ukuran area plot saat ini dan menyesuaikan jendela agar label-labelnya memiliki ruang yang cukup.

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

Fungsi reset() mengembalikan pengaturan plot ke default, termasuk rasio aspek.
Plot 2D di Euler

EMT Math Toolbox memiliki plot 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d untuk membuat plot dari fungsi dan data.

Kamu juga bisa membuat plot di Maxima menggunakan Gnuplot atau di Python menggunakan Matplotlib.

Euler dapat membuat plot 2D dari:

- ekspresi

- fungsi, variabel, atau kurva yang diparameterisasi,

- vektor nilai x-y,

- kumpulan titik di bidang,

- kurva implisit dengan level atau wilayah level,

- fungsi kompleks.

Gaya plot termasuk berbagai gaya untuk garis dan titik, plot batang, dan plot berbayangan.
Plot Ekspresi atau Variabel

Satu ekspresi dalam "x" (misalnya "4*x^2") atau nama fungsi (misalnya "f") akan menghasilkan grafik dari fungsi tersebut.

Berikut adalah contoh paling dasar, yang menggunakan rentang default dan mengatur rentang y yang sesuai untuk menyesuaikan plot fungsi tersebut.

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

```
>plot2d("x^2"):
>aspect(1.5); plot2d("x^3-x"):
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```

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

Rentang plot diatur dengan parameter-parameter berikut:

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

```
>plot2d("x^3-x",-1,2):
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi):
```

Sebagai alternatif untuk titik dua ":", kamu bisa menggunakan perintah insimg(lines), yang menyisipkan plot dan menggunakan sejumlah baris teks yang ditentukan.

Dalam opsi, plot dapat diatur untuk muncul:
- di jendela terpisah yang bisa diubah ukurannya,
- di jendela notebook.

Gaya plot tambahan dapat dicapai dengan menggunakan perintah plot khusus.

Jika plot tersembunyi, tekan tombol tabulator untuk melihatnya.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh ini, kita memplot x^1 hingga x^4 ke dalam 4 bagian jendela. Perintah figure(0) mengatur ulang jendela ke pengaturan default.

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

Dalam plot2d(), tersedia gaya alternatif dengan menggunakan grid=x. Untuk memberikan gambaran, kami menunjukkan berbagai gaya grid dalam satu gambar (lihat perintah figure() di bawah). Gaya grid=0 tidak termasuk. Gaya ini tidak menampilkan 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 pada plot2d() adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot tersebut.

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

Pada 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-gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks disimpan di direktori yang sama dengan notebook, secara default dalam subdirektori bernama "images". Gambar-gambar ini juga digunakan saat mengekspor ke HTML.

Kamu bisa dengan mudah menandai gambar apa pun dan menyalinnya ke clipboard dengan menekan Ctrl-C. Tentu saja, kamu juga bisa mengekspor grafik saat ini dengan fungsi-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 perlu dilakukan dalam kasus-kasus yang jarang.

```
>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 ditentukan untuk x<=0. Fungsi plot2d menangkap kesalahan ini, dan mulai membuat plot segera setelah fungsinya ditentukan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN di luar jangkauan definisinya.

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

Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan spasi 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 memerlukan lebih banyak ruang untuk label y, panggil shrinkwindow() dengan parameter 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 di file "plot.e", yang dimuat di awal program.

Berikut beberapa contoh penggunaan suatu fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda bisa meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan kumpulan 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 ini ringkasan fungsi yang diterima
- ekspresi atau ekspresi simbolik di x
- fungsi atau fungsi simbolik dengan nama "f"
- fungsi simbolik hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, namanya saja yang berfungsi.

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

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

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

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

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

$$-x \\ E \quad \sin(x)$$

```
>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: Defaultnya 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 muda, kuning
- rgb(merah,hijau,biru): parameternya real di [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)
>insimg (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 metodenya adalah menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, kecuali panggilan pertama. Kami telah menggunakan fitur ini pada 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):
```

Kita tambahkan titik perpotongan dengan label (pada posisi "cl" untuk kiri tengah), dan masukkan hasilnya ke dalam buku catatan. Kami juga menambahkan judul pada plot.

```
>plot2d(["cos(x)","x"],r=1.1,cx=0.5,cy=0.5, ...
> color=[black,blue],style=["-","."], ...
> grid=1);
```

```
  Illegal parameter after named parameter!
  Error in:
  ... .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, kita memplot fungsi sin(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung perluasan ini menggunakan Maxima melalui ekspresi simbolik.
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke plot2d(). Yang kedua dan ketiga memiliki kumpulan tanda >add, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsinya.

```
>$taylor(sin(x)/x,x,0,4)
>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 Polinomial Bernstein.
lateks: 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;
```

Cara kedua adalah dengan menggunakan pasangan matriks bernilai x dan matriks bernilai y yang berukuran sama.
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pendahuluan 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 susunan warna, susunan gaya, dan susunan 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
```

$$\left[(1-x)^{10},\ 10\,(1-x)^{9}\,x,\ 45\,(1-x)^{8}\,x^{2},\ 120\,(1-x)^{7}\,x^{3},\ 210\,(1-x)^{6}\,x^{4},\ 252\,(1-x)^{5}\,x^{5},\ 210\,(1-x)^{4}\,x^{6},\ 120\,(1-x)^{3}\,x^{7},\ 45\,(1-x)^{2}\,x^{8},\ 10\,(1-x)\,x^{9},\ x^{10}\right]$$

```
>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 suatu ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi tersebut akan diplot ke dalam satu plot.

Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika array warna ditambahkan maka 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 ini 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"):
```

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

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

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

Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks f(x,a) merupakan 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)):
```

Dekorasi sederhana pun bisa

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

Perintah label akan memplot ke plot saat ini pada koordinat plot (x,y). Hal ini memerlukan argumen posisional.

```
>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 ini berlabuh di kanan atas secara default, tetapi >kiri berlabuh di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar berada di pojok kanan atas kotak, dan angkanya merupakan pecahan dari ukuran jendela grafis. Lebarnya otomatis.

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

Pada 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. Masih banyak lagi plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

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

Fitur serupa adalah fungsi textbox().

Lebarnya secara default adalah lebar maksimal 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 dengan 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, penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil latex() sebelum loop satu kali dan menggunakan hasilnya (gambar dalam matriks RGB).

Pada 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, kita menginginkan spasi dan label teks yang tidak konformal pada sumbu x. Kita bisa menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.

Cara termudah adalah membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Pada 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 fungsinya juga bisa 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. Tapi untuk menunjukkan vektorisasi itu
berguna, kita menambahkan beberapa poin penting 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 bisa menggunakannya
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 suatu fungsi atau ekspresi, parameter >pengguna memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna bisa

- 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 aslinya.

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 aktivitas mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse digerakkan 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. Misalnya, kita melakukan interpolasi pada 5 titik dengan polinomial. Fungsi tersebut harus diplot ke dalam area plot yang tetap.

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

  endfunction
```

```
>
```

```
    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 nilainya secara global.

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

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

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 nilainya secara global.

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

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

Kemudian kita memerlukan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional garis judul. Ada penggeser interaktif, yang dapat menetapkan 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 menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor berderajat 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 kita izinkan derajat n bervariasi dari 0 hingga 20 dalam 20 perhentian. 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 fungsinya. Pengguna dapat menggambar jendela plot, meninggalkan jejak titik.

```
>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 diubah. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk menyetel ulang ke gaya default, gunakan reset().

```
>aspect(); insimg;
>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 insid
> 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
> figure(9); plot2d("x^3-x",grid=8); ... // axes only, finer
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks
> figure(11); plot2d("x^3-x",grid=10); ...// no ticks, axes only
> figure(0):
```

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

Jika Anda menginginkan tanda centang Anda 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 bagian tengah bawah. Ini menetapkan posisi label relatif terhadap koordinat plot.

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

$$x^3 - x$$

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

Ada juga kotak teks.

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

```
  Closing bracket missing in function call!
  Error in:
  ... p(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 lanjut, sumbu x dan sumbu y dapat dilakukan secara manual.

Perintah fullwindow() memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk menyetel 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 sumbunya berada 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, maka data tersebut akan digunakan sebagai koordinat x dan y pada suatu kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Alternatifnya, >persegi dapat diatur untuk mempertahankan rasio aspek persegi.

Merencanakan ekspresi hanyalah singkatan dari 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 dan titik campuran gunakan ">addpoints".

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 poin. Gunakan points=true untuk ini. Plotnya berfungsi seperti poligon, tetapi hanya menggambar sudutnya saja.

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

Untuk memplot kumpulan titik, gunakan >titik. Jika warna merupakan vektor warna, masing-masing titik mendapat warna berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada 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 sebenarnya berbentuk poligon. Kita juga dapat memplot kurva atau kurva terisi.

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

Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional, mencegah menggambar batas untuk semua gaya kecuali gaya 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 kita 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 septagon 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 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 pembuatan tabel fungsi dengan mudah.

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

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

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

Vektor interval diplot terhadap nilai x sebagai wilayah terisi antara nilai interval yang lebih rendah dan lebih tinggi.

Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga dapat 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 pada bidang. Gaya isiannya 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
```

Kita juga dapat mengisi rentang nilai seperti

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

```
>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 sebuah kurva. Jika x diurutkan, kurva tersebut merupakan grafik suatu fungsi.

Dalam contoh berikut, kita 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 adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih jelasnya).

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

Sebagai alternatif, 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):
```

Pada contoh berikutnya, kita memplot kurvanya

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

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

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

Matriks bilangan kompleks secara otomatis akan diplot sebagai kisi-kisi pada bidang kompleks.

Pada contoh berikut, kita 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);...
>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):
```

Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian nyata dan bagian imajiner.

Dalam contoh, kita memplot lingkaran satuan dengan

lateks: \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 normal 0-1 menghasilkan jalan acak.

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

Penggunaan dua baris menunjukkan jalan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
>columnsplot(cumsum(random(10)),style="/",color=blue):
>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() mengatur 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. Batangnya akan memanjang dari x[i] hingga x[i+1] dengan nilai y[i]. Jika x berukuran sama dengan y, maka x 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 (batang=1) dan histogram (histogram=1) dapat diberikan 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 >even 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, terdapat parameter distribution=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 bilangan bulat.

```
>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. Plot kotak menunjukkan kuartil distribusi ini dan banyak outlier. Menurut definisinya, outlier dalam plot kotak adalah data yang melebihi 1,5 kali rentang 50% tengah plot.

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

Fungsi Implisit

Plot implisit menunjukkan penyelesaian garis level f(x,y)=level, dengan "level" dapat berupa nilai tunggal atau vektor nilai. Jika level = "auto", akan ada garis level nc, yang akan tersebar 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 parameter x dan y, atau alternatifnya, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

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

dari fungsi apa pun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisitnya pada bidang. Parameter c adalah level=c, dimana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi 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 juga berfungsi untuk plot data. Namun 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):
>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):
```

Dimungkinkan juga untuk mengisi set

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

dengan rentang level.

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. Maka level harus berupa matriks interval level 2xn, di mana baris pertama berisi awal dan baris kedua berisi akhir setiap interval. Alternatifnya, 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

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

Hal 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 tanda centang kotak atau sumbu di sini. Selain itu, kami menggunakan jendela penuh untuk plot-nya.

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

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

Terkadang, Anda mungkin ingin merencanakan sesuatu yang plot2d tidak bisa lakukan, tapi hampir.

Dalam fungsi berikut, kita membuat 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 dalam jendela plot. setplot(a,b,c,d) menyetel jendela ini.

Fungsi wait(0) memaksa plot muncul di jendela grafis. Jika tidak, pengundian ulang akan dilakukan 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 tombol apa saja 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 menggunakan skala logaritma di y dengan logplot=1, atau menggunakan skala logaritma di x dan y dengan logplot=2, atau di 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);
>hold off;
```

Rujukan Lengkap Fungsi plot2d()

fungsi plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, ..
logplot, kisi, bingkai, warna bingkai, kotak, warna, ketebalan, gaya, ..
otomatis, tambah, pengguna, delta, titik, titik tambahan, gaya titik, batang, histogram, ..
distribusi, genap, langkah, milik, adaptif, rona, level, kontur, ..
nc, terisi, warna isi, garis besar, judul, xl, yl, peta, warna kontur, ..
lebar kontur, kutu, margin, kliping, cx, cy, insimg, spektral, ..
cgrid, vertikal, lebih kecil, dl, niveau, level)

Fungsi plot serbaguna untuk plot dalam bidang (plot 2D). Fungsi ini dapat dilakukan
plot fungsi satu variabel, plot data, kurva pada bidang, plot batang, grid
bilangan kompleks, dan plot implisit fungsi dua variabel.

Parameters

x,y : equations, functions or data vectors
a,b,c,d : Plot area (default a=-2,b=2)
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r

```
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
```

xmin,xmax : range of the parameter for curves
auto : Determine y-range automatically (default)
square : if true, try to keep square x-y-ranges
n : number of intervals (default is adaptive)
grid : 0 = no grid and labels,

```
1 = axis only,
2 = normal grid (see below for the number of grid lines)
3 = inside axis
4 = no grid
5 = full grid including margin
6 = ticks at the frame
7 = axis only
8 = axis only, sub-ticks
```

frame : 0 = no frame
framecolor: color of the frame and the grid
margin : number between 0 and 0.4 for the margin around the plot
color : Color of curves. If this is a vector of colors,it will be used for each row of a matrix of plots. In the case of point plots, it should be a column vector. If a row vector or a full matrix of colors is used for point plots, it will be used for each data point.
thickness : line thickness for curves
This value can be smaller than 1 for very thin lines.
style : Plot style for lines, markers, and fills.

```
For points use
"[]", "<>", ".", "..", "...",
"*", "+", "|", "-", "o"
"[]#", "<>#", "o#" (filled shapes)
"[]w", "<>w", "ow" (non-transparent)
For lines use
"-", "--", "-.", ".", ".-.", "-.-", "->"
For filled polygons or bar plots use
"#", "#O", "O", "/", "\", "\/",
"+", "|", "-", "t"
```

points : plot single points instead of line segments
addpoints : if true, plots line segments and points
add : add the plot to the existing plot
user : enable user interaction for functions
delta : step size for user interaction
bar : bar plot (x are the interval bounds, y the interval values)
histogram : plots the frequencies of x in n subintervals
distribution=n : plots the distribution of x with n subintervals
even : use inter values for automatic histograms.
steps : plots the function as a step function (steps=1,2)
adaptive : use adaptive plots (n is the minimal number of steps)
level : plot level lines of an implicit function of two variables
outline : draws boundary of level ranges.

If the level value is a 2xn matrix, ranges of levels will be drawn in the color using the given fill style. If outline is true, it

will be drawn in the contour color. Using this feature, regions of
f(x,y) between limits can be marked.

hue : add hue color to the level plot to indicate the function value
contour : Use level plot with automatic levels
nc : number of automatic level lines
title : plot title (default "")
xl, yl : labels for the x- and y-axis
smaller : if >0, there will be more space to the left for labels.
vertical :

```
Mengaktifkan atau menonaktifkan label vertikal.Ini   mengubah
```

variabel global memberi label vertikal secara lokal untuk satu plot. Nilai 1 hanya ditetapkan secara vertikal teks, nilai 2 menggunakan label numerik vertikal pada sumbu y.
filled : fill the plot of a curve
fillcolor : fill color for bar and filled curves
outline : boundary for filled polygons
logplot : set logarithmic plots

```
1 = logplot in y,
2 = logplot in xy,
3 = logplot in x
```

own :
Sebuah string, yang menunjuk ke rutinitas plotnya sendiri. Dengan> pengguna, Anda mendapatkan interaksi pengguna yang sama seperti di plot2d. Kisarannya akan ditentukan sebelum setiap panggilan ke fungsi Anda.
maps : map expressions (0 is faster), functions are always mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping : toggles the clipping (default is true)
title :
Ini dapat digunakan untuk mendeskripsikan plot. Judulnya akan muncul di atas plotnya. Selain itu, label untuk sumbu x dan y dapat ditambahkan xl="string" atau yl="string". Label lain dapat ditambahkan dengan fungsi label() atau labelbox(). Judulnya bisa berupa unicode string atau gambar rumus Lateks.
cgrid :
Menentukan jumlah garis grid untuk plot grid yang kompleks. Harus berupa pembagi ukuran matriks dikurangi 1 (jumlah subinterval). cgrid dapat berupa vektor [cx,cy].

Ringkasan

Fungsinya dapat diplot
- ekspresi, kumpulan panggilan atau fungsi dari satu variabel,
- kurva parametrik,
- x data versus y data,
- fungsi implisit,
- plot batang,
- jaringan kompleks,
- poligon.

Jika fungsi atau ekspresi untuk xv diberikan, plot2d() akan menghitung nilai dalam rentang tertentu menggunakan fungsi atau ekspresi. Itu ekspresi harus berupa ekspresi dalam variabel x. Kisarannya harus ditentukan dalam parameter a dan b kecuali rentang default [-2,2] harus digunakan. Rentang y akan dihitung secara otomatis, kecuali c dan d ditentukan, atau radius r, yang menghasilkan rentang [-r,r] untuk x dan y. Untuk plot fungsi, plot2d akan menggunakan evaluasi adaptif fungsi secara default. Untuk mempercepat plot untuk fungsi yang rumit, nonaktifkan ini dengan <adaptive, dan secara opsional mengurangi jumlah interval n. Selain itu, plot2d() akan secara default menggunakan pemetaan. Yaitu, ini akan menghitung elemen plot untuk

elemen. Jika ekspresi atau fungsi Anda dapat menangani a vektor x, Anda dapat menonaktifkannya dengan <maps untuk evaluasi lebih cepat.

Perhatikan bahwa plot adaptif selalu dihitung elemen demi elemen. Jika fungsi atau ekspresi untuk xv dan yv ditentukan, plot2d() akan menghitung kurva dengan nilai xv sebagai koordinat x dan nilai yv sebagai koordinat y. Dalam hal ini, seharusnya ada kisaran ditentukan untuk parameter menggunakan xmin, xmax. Ekspresi terkandung dalam string harus selalu berupa ekspresi dalam variabel parameter x.

Contoh Soal dan penyelesaiannya

```
>clg; // clear screen
>window(0,0,2045,2045); // use all of the window
>setplot(0,2,0,2); // set plot coordinates
>hold on; // start overwrite mode
>n=150; X=random(n,4); Y=random(n,4); // 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
```

PLOT3D

* Menggambar Plot 3D dengan EMT

Ini adalah pengantar plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

Euler menggambar fungsi tersebut menggunakan algoritma penyortiran untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah titik asal x=y=z=0, tetapi sudut=0° terlihat dari arah sumbu y. Sudut pandang dan tinggi dapat diubah.

Euler dapat memplot

- permukaan dengan bayangan dan garis datar atau rentang datar,
- awan titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D suatu fungsi menggunakan plot3d. Cara termudah adalah memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

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

Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang. Temukan rumusnya.

Fungsi Dua Variabel

Untuk grafik fungsi, gunakan

- ekspresi sederhana dalam x dan y,
- nama fungsi dua variabel
- atau matriks data.

Defaultnya adalah kisi kawat yang terisi dengan warna berbeda di kedua sisinya. Perhatikan bahwa jumlah interval kisi default adalah 10, tetapi plot menggunakan jumlah persegi panjang 40x40 default untuk membuat permukaan. Ini dapat diubah.

- n=40, n=[40,40]: jumlah garis kisi di setiap arah
- kisi=10, kisi=[10,10]: jumlah garis kisi di setiap arah.

Kami menggunakan default n=40 dan kisi=10.

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

Interaksi pengguna dimungkinkan dengan parameter >user. Pengguna dapat menekan tombol berikut.

- kiri, kanan, atas, bawah: mengubah sudut pandang
- +, -: memperbesar atau memperkecil
- a: menghasilkan anaglif (lihat di bawah)
- l: mengubah arah sumber cahaya (lihat di bawah)
- spasi: mengatur ulang ke default
- return: mengakhiri interaksi

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

Rentang plot untuk fungsi dapat ditentukan dengan

- a,b: rentang x
- c,d: rentang y
- r: persegi simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala ke nilai fungsi (default adalah <fscale).
scale: angka atau vektor 1x2 untuk diskalakan ke arah x dan y.
frame: jenis frame (default 1).

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

Tampilan dapat diubah dengan berbagai cara.

- jarak: jarak pandang ke plot.
- perbesaran: nilai perbesaran.
- sudut: sudut ke sumbu y negatif dalam radian.
- tinggi: tinggi tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi view(). Fungsi ini mengembalikan parameter dalam urutan di atas.

```
>view
```

```
  [5,  2.6,  2,  0.4]
```

Jarak yang lebih dekat membutuhkan zoom yang lebih sedikit. Efeknya lebih seperti lensa sudut lebar.

Dalam contoh berikut, sudut=0 dan tinggi=0 terlihat 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):
```

Plot selalu mengarah ke tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

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

Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label merujuk pada ukuran sebenarnya.

Jika Anda menonaktifkannya dengan scale=false, Anda perlu berhati-hati agar plot tetap sesuai dengan jendela plot, dengan mengubah jarak tampilan 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):
```

Plot polar juga tersedia. Parameter polar=true menggambar plot polar. Fungsi tersebut harus tetap berupa fungsi x dan y. Parameter "fscale" menskalakan fungsi dengan skalanya sendiri. Jika tidak, fungsi tersebut 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):
>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):
```

Parameter rotate memutar fungsi dalam x di sekitar sumbu x.

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

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```

Berikut adalah plot dengan tiga fungsi.

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

Plot Kontur

Untuk plot, Euler menambahkan garis kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona warna spektral. Euler dapat menggambar tinggi fungsi pada plot dengan bayangan. Dalam semua plot 3D, Euler dapat menghasilkan anaglif merah/sian.

- >hue: Mengaktifkan bayangan terang alih-alih kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Vektor nilai untuk garis kontur.

Defaultnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan kisi yang lebih halus untuk titik 100x100, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
> >contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```

Shading default menggunakan warna abu-abu. Namun, rentang warna spektral juga tersedia.
- >spectral: Menggunakan skema spektral default
- color=...: Menggunakan warna khusus atau skema spektral

Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat halus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```

Alih-alih garis level otomatis, kita juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis, bukan rentang level.

```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```

Dalam plot berikut, kami menggunakan dua pita level yang sangat lebar dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.
Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>  >spectral,angle=30°,grid=10,contourcolor=gray):
```

Dalam contoh berikut, kami memplot himpunan, di mana

$$f(x,y) = x^y - y^x = 0$$

Kami menggunakan satu garis tipis untuk garis datar.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```

Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```

Berikut ini beberapa gaya lainnya. Kami selalu menonaktifkan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1);  ...
>  plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2);  ...
>  plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3);  ...
>  plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4);  ...
>  plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```

Ada beberapa skema spektral lain, yang diberi nomor dari 1 hingga 9. Namun, Anda juga dapat menggunakan color=value, di mana value

- spektral: untuk rentang dari biru hingga merah
- putih: untuk rentang yang lebih redup
- kuning biru, ungu hijau, biru kuning, hijau merah
- biru kuning, hijau ungu, kuning biru, merah hijau

```
>figure(3,3); ...
>for i=1:9;  ...
>  figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4);  ...
>end; ...
>figure(0):
```

Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Sumber cahaya juga dapat diatur dengan parameter.

- light: arah cahaya
- amb: cahaya sekitar antara 0 dan 1

Perlu dicatat bahwa program tidak membuat perbedaan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda memerlukan Povray.

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press l and cursor keys (return to exit)"):
```

Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
>  zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```

Warna 0 memberikan efek pelangi khusus.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```

Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```

Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d mencakup plot implisit. Plot ini menunjukkan himpunan nol dari suatu fungsi dalam tiga variabel.
Solusi dari

$$f(x,y,z)=0$$

dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, x-z, dan y-z.
- implisit=1: potongan sejajar dengan bidang y-z
- implisit=2: potongan sejajar dengan bidang x-z
- implisit=4: potongan sejajar dengan bidang x-y

Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh ini, kami memplot

$$M=\{(x,y,z):x^2+y^3+zy=1\}$$

```
>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):
```

Membuat Plot Data 3D

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, kami berasumsi bahwa (t,s) berjalan melalui kisi persegi. Hasilnya, Anda dapat membuat plot gambar persegi panjang di ruang angkasa.
Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.
Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar, kami dapat menandai wilayah, dalam kasus kami wilayah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```

Berikut adalah contoh, yang merupakan grafik suatu fungsi.

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

Namun, kita dapat membuat berbagai macam permukaan. Berikut ini adalah permukaan yang sama sebagai fungsi
lateks: x = y \, z

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```

With more effort, we can produce many surfaces.
In the following example we make a shaded view of a distorted ball. The usual coordinates for the ball are

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

with

$$0 \le t \le 2\pi, \quad \frac{-\pi}{2} \le s \le \frac{\pi}{2}.$$

We distored this with a factor

$$d(t,s) = \frac{\cos(4t)+\cos(8s)}{4}.$$

```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
>  light=[1,0,1],frame=0,zoom=5):
```

Tentu saja, titik awan juga memungkinkan. Untuk memplot data titik di ruang, kita memerlukan tiga vektor untuk koordinat titik.
Gayanya sama seperti di plot2d dengan points=true;

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

Anda juga dapat memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung titik-titik kurva terlebih dahulu. Untuk kurva dalam bidang, kami menggunakan urutan koordinat dan parameter wire=true.

```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3,wirecolor=blue):
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```

EMT juga dapat membuat grafik dalam mode anaglif. Untuk melihat grafik tersebut, Anda memerlukan kacamata merah/sian.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```

Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan tinggi fungsi.

```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```

Euler juga dapat memplot permukaan berparameter, ketika parameternya adalah nilai x, y, dan z dari gambar kotak persegi panjang di ruang tersebut.

Untuk demo berikut, kami menyiapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter tersebut.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```

Berikut adalah contoh yang lebih rumit, yang tampak megah dengan kaca merah/cyan.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)';  ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
> z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```

Plot Statistik

Plot batang juga dimungkinkan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

Dalam contoh, pertama-tama kita menghitung nilai. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```

Dimungkinkan untuk membagi bidang permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20):
```

Jika memuat atau membuat matriks data M dari sebuah file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan >zscale. Ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```

Permukaan Benda Putar

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

Kita ingin memutar kurva jantung di sekitar sumbu y. Berikut ini adalah ekspresi yang mendefinisikan jantung:

$$f(x,y) = (x^2+y^2-1)^3 - x^2.y^3.$$

Selanjutnya kita tetapkan

$$x = r.cos(a), \quad y = r.sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang memecahkan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang diputar sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t);  ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi yang menggambarkan objek tersebut.

```
>function f(x,y,z) ...
```

```
  r=x^2+y^2;
  return (r+z^2-1)^3-r*z^3;
   endfunction
```

```
>plot3d("f(x,y,z)", ...
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
>implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```

Plot 3D Khusus

Fungsi plot3d memang bagus, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda dapat memperoleh plot berbingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukanlah program 3D, ia dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan parabola dan garis singgungnya.

```
>function myplot ...
```

```
    y=-1:0.01:1; x=(-1:0.01:1)';
    plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
      hues=0.5,>contour,color=orange);
    h=holding(1);
    plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
    holding(h);
  endfunction
```

Sekarang framedplot() menyediakan bingkai dan mengatur tampilan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
>  center=[0,0,-0.7],zoom=3):
```

Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() menetapkan jendela ke fullwindow() secara default, tetapi plotcontourplane() mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
    zoom(2);
    wi=fullwindow();
    plotcontourplane(x,y,z,level="auto",<scale);
    plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
    window(wi);
    reset();
  endfunction
```

```
>myplot(x,y,z):
```

Animasi

Euler dapat menggunakan bingkai untuk melakukan pra-komputasi animasi.

Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil addpage() untuk setiap plot baru. Terakhir, fungsi ini menganimasikan plot tersebut.

Silakan pelajari sumber rotate untuk melihat detail selengkapnya.

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

Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat membuat file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari http://www.povray.org/, dan meletakkan subdirektori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler membuat file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori default adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray membuat file PNG, yang dapat dimuat oleh Euler ke dalam buku catatan. Untuk membersihkan file-file ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik fungsi f(x,y), atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam buku catatan Euler.

Selain pov3d(), ada banyak fungsi, yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file adegan. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan mulai, dan PNG akan dimasukkan ke dalam buku catatan Euler.

Fungsi objek memiliki parameter yang disebut "look", yang memerlukan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan sumbu x, y, z dalam arah kanan.
Anda perlu memuat berkas povray.

```
>load povray;
```

Pastikan direktori bin Povray ada di jalur tersebut. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
  C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk kesan pertama, kami membuat fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melakukan ray tracing pada file ini.

Jika Anda menjalankan perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya apakah Anda ingin mengizinkan file exe untuk berjalan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengakui dialog awal Povray.

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

Kita dapat membuat fungsi tersebut transparan dan menambahkan penyelesaian lainnya. Kita juga dapat menambahkan garis level pada plot fungsi.

```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...
>  look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi secara manual.

Kami memplot himpunan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
>  angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
>  <fscale,zoom=3.8);
```

Membuat Plot dengan Koordinat

Alih-alih menggunakan fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita memerlukan tiga matriks untuk menentukan objek.

Dalam contoh ini, kita memutar fungsi di sekitar sumbu z.

```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```

Dalam contoh berikut, kami memplot gelombang yang diredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3;  ...
>pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
>  w=500,h=300);
```

Dengan metode shading Povray yang canggih, hanya sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya pada batas dan bayangan, triknya mungkin menjadi jelas.

Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$x^2\,y^3$$

Persamaan permukaannya adalah [x,y,Z]. Kita hitung dua turunan x dan y dari persamaan ini dan ambil perkalian silang sebagai normalnya.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

Kami mendefinisikan normal sebagai perkalian silang turunan-turunan ini dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\left[-2\,x\,y^3,\ -3\,x^2\,y^2,\ 1\right]$$

Kami hanya menggunakan 25 poin.

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
>  xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```

Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari simpul ini dalam contoh-contohnya.

Lihat: Contoh\Simpul Trefoil | Simpul Trefoil

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan ke x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang merupakan perkalian silang dari dua turunan.

```
>dn &= crossproduct(dx,dy);
```

Sekarang mari kita evaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

Vektor normal adalah evaluasi ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks untuk ini adalah &"ekspresi"(parameter). Ini adalah alternatif untuk metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
>  <shadow,look=povlook(blue), ...
>  xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```

Kita juga dapat membuat grid dalam 3D.

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```

Dengan povgrid(), kurva dimungkinkan.

```
>povstart(center=[0,0,1],zoom=3.6); ...
>t=linspace(0,2,1000); r=exp(-t); ...
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
>writeln(povgrid(x,y,z,povlook(red))); ...
>writeAxis(0,2,axis=3); ...
>povend();
```

Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke berkas Povray.

Kami memulai output dengan povstart().

```
>povstart(zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam string di Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

Rangkaian tersebut berisi kode Povray, yang tidak perlu kita pahami saat itu.

```
>c2
```

```
  cylinder { <0,0,-1>, <0,0,1>, 1
   texture { pigment { color rgb <0.941176,0.941176,0.392157> }  }
   finish { ambient 0.2 }
   }
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Hal itu dilakukan oleh povlook(), yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

```
   texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> }  }
   finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek persimpangan dan menulis hasilnya ke berkas.

```
>writeln(povintersection([c1,c2,c3]));
```

Persimpangan tiga silinder sulit dibayangkan, jika Anda belum pernah melihatnya sebelumnya.

```
>povend;
```

Fungsi-fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan bagaimana Euler menangani objek-objek Povray sederhana. Fungsi povbox() mengembalikan string yang berisi koordinat kotak, tekstur, dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());
>function fractal (x,y,z,h,n) ...
```

```
   if n==1 then writeln(onebox(x,y,z,h));
   else
     h=h/3;
     fractal(x,y,z,h,n-1);
     fractal(x+2*h,y,z,h,n-1);
     fractal(x,y+2*h,z,h,n-1);
     fractal(x,y,z+2*h,h,n-1);
```

```
    fractal(x+2*h,y+2*h,z,h,n-1);
    fractal(x+2*h,y,z+2*h,h,n-1);
    fractal(x,y+2*h,z+2*h,h,n-1);
    fractal(x+2*h,y+2*h,z+2*h,h,n-1);
    fractal(x+h,y+h,z+h,h,n-1);
  endif;
endfunction
```

```
>povstart(fade=10,<shadow);
>fractal(-1,-1,-1,2,4);
>povend();
```

Perbedaan memungkinkan pemisahan satu objek dari objek lainnya. Seperti halnya persimpangan, ada beberapa objek CSG dari Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan objek dalam Povray, alih-alih menggunakan string dalam Euler. Definisi langsung ditulis ke berkas.
Koordinat kotak -1 berarti [-1,-1,-1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kita buat kubus kedua, lalu putar dan ubah skalanya sedikit.

```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
>  rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Lalu kita ambil selisih kedua benda tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
>writeAxis(-1.2,1.2,axis=2); ...
>writeAxis(-1.2,1.2,axis=4); ...
>povend();
```

Fungsi Implisit

Povray dapat memplot himpunan di mana f(x,y,z)=0, sama seperti parameter implisit dalam plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)=d$$

```
>povstart(angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
>writeln(povsurface("(pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*(pow(pow(y,2)+p
>povend();
```

```
  Error : Povray error!

  Error generated by error() command

  povray:
      error("Povray error!");
  Try "trace errors" to inspect local variables after errors.
  povend:
      povray(file,w,h,aspect,exit);
```

```
>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
>povend();
>povstart(angle=70°,height=50°,zoom=4);
```

Buat permukaan implisit. Perhatikan sintaksis yang berbeda dalam ekspresi.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
>writeAxes(); ...
>povend();
```

Objek Mesh

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy dalam kondisi x+y=1 dan menunjukkan sentuhan tangensial garis-garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukan ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti pov3d().

Berikut ini mendefinisikan objek mesh, dan langsung menuliskannya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita definisikan dua cakram, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tuliskan permukaan dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan dua titik potongnya.

```
>writeln(povintersection([mesh,cl],povlook(red)); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tuliskan titik maksimumnya.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesaikan.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

Anaglif dalam Povray

Untuk menghasilkan anaglif untuk kacamata merah/sian, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ia menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph(). Tentu saja, Anda memerlukan kacamata merah/sian untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki sakelar sederhana untuk menghasilkan anaglif.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
>  center=[0,0,0.5],zoom=3.5);
```

Jika Anda membuat suatu pemandangan dengan objek, Anda perlu memasukkan pembuatan pemandangan tersebut ke dalam suatu fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
```

```
    s=povsphere(povc,1);
    cl=povcylinder(-povz,povz,0.5);
    clx=povobject(cl,rotate=xrotate(90°));
    cly=povobject(cl,rotate=yrotate(90°));
    c=povbox([-1,-1,0],1);
    un=povunion([cl,clx,cly,c]);
    obj=povdifference(s,un,povlook(red));
    writeln(obj);
    writeAxes();
  endfunction
```

Fungsi povanaglyph() melakukan semua ini. Parameternya seperti pada povstart() dan povend() yang digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```

Menentukan Objek Sendiri

Antarmuka povray Euler berisi banyak objek. Namun, Anda tidak terbatas pada objek-objek ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau objek yang sama sekali baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi, kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat di titik asal.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {"+r1+","+r2+look+"}";
  endfunction
```

Inilah torus pertama kita.

```
>t1=povdonat(0.8,0.2)
```

```
  torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
  object { torus {0.8,0.2}
   rotate 90 *x
   translate <0.8,0,0>
   }
```

Sekarang kita tempatkan objek-objek ini ke dalam sebuah scene. Untuk tampilannya, kita gunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

>povend();
memanggil program Povray. Namun, jika terjadi kesalahan, program tersebut tidak menampilkan kesalahan tersebut. Oleh karena itu, Anda harus menggunakan
>povend(<exit);
jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```

Berikut adalah contoh yang lebih rinci. Kami memecahkan

$$Ax \le b, \quad x \ge 0, \quad c.x \to \text{Max}.$$

dan menunjukkan titik-titik yang layak dan titik-titik optimum dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini punya solusi.

```
>x=simplex(A,b,c,>max,>check)'
```

```
  [0,  1,  0.5]
```

Ya, benar.
Berikutnya kita mendefinisikan dua objek. Yang pertama adalah bidang datar

$$a \cdot x \le b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
  endfunction
```

Kemudian kita mendefinisikan irisan semua ruang setengah dan sebuah kubus.

```
>function adm (A, b, r, look="") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
  endfunction
```

Sekarang, kita dapat merencanakan adegannya.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut ini adalah lingkaran di sekitar titik optimum.

```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>  povlook(red,0.9)));
```

Dan kesalahan dalam arah yang optimum.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kita menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5°)); ...
>povend();
```

Contoh Lainnya

Anda dapat menemukan beberapa contoh lainnya untuk Povray di Euler dalam berkas berikut.

Lihat: Contoh/Bola Dandelin
Lihat: Contoh/Donat Math
Lihat: Contoh/Simpul Trefoil
Lihat: Contoh/Optimasi dengan Penskalaan Afinitas

Contoh dari Nadzwa

Nadzwa adalah seorang peneliti yang sedang mempelajari distribusi suhu pada sebuah permukaan datar yang berbentuk persegi. Permukaan ini memiliki koordinat x dari y yang berkisar dari -2 hingga 2 dengan selisih 0.1 dan y sebagai transpose dari x. Nadzwa ingin memvisualisasikan distribusi suhu ini berdasarkan fungsi matematika yang menggambarkan variasi suhu pada setiap titik, yaitu

$$z = x^2 + y^2$$

dimana z mewakili suhu di titik dengan koordinat (x,y).
Selanjutnya Nadzwa ingin melakukan modifikasi pada x dan y dengan mengurangi masing-masing nilai sebesar 0.05.

```
>x=-2:0.5:2; y=x'; z=x^2+y^2;...
>xa=(x|2.1)-0.5; ya=(y_2.1)-0.5;...
>plot3d(xa,ya,z,bar=true):
```

Kalkulus EMT

```
>
```

Materi Kalkulus mencakup di antaranya:
- Fungsi (fungsi aljabar, trigonometri, eksponensial,logaritma,

```
komposisi fungsi)
```

- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

Mendefiniskan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:
- Menggunakan format nama_fungsi := rumus fungsi (untuk fungsi

```
numerik),
```

- Menggunakan format nama_fungsi &= rumus fungsi (untuk fungsi

```
simbolik, namun dapat dihitung secara numerik),
```

- Menggunakan format nama_fungsi &&= rumus fungsi (untuk fungsi

```
simbolik murni, tidak dapat dihitung langsung),
```

- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).
Berikut adalah adalah beberapa contoh cara mendefinisikan fungsi.

Definisi Fungsi Aljabar

fungsi matematika yang didefinisikan menggunakan operasi aljabar dasar seperti penjumlahan, pengurangan, perkalian, pembagian, serta pangkat rasional.
Misalkan kita punya fungsi aljabar:

$$f(x) = 2x + 3$$

cari f(0), f(2), f(3) dan buatkan grafik 2d

```
>function f(x) := 2*x+3 // fungsi numerik
>f(0)
```

```
  3
```

```
>f(2)
```

```
  7
```

```
>f(3)
```

```
  9
```

```
>plot2d("2*x+3"):
```

Berikutnya kita definisikan fungsi:

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

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

```
  0
```

```
>g(0)
```

```
  0
```

dapat dilihat bahwa x=0 dan x=3 membuat nilai g(x)=0, artinya x=0 dan x=3 adalah akar-akar dari persamaan pada pembilang di fungsi g(x).

```
>plot2d("sqrt(x^2-3*x)/(x+1)"):
```

latihan Soal

Misalkan kita punya fungsi aljabar:

$$g(x) = \frac{\sqrt{x^2 - 5x}}{2x - 3}$$

cari g(0), g(5) dan buatkan plot 2d

```
>function g(x) := sqrt(x^2-5x)/(2x-3)
>g(0), g(5)
```

```
  0
  0
```

```
>plot2d("sqrt(x^2-5x)/(2x-3)"):
>&load(fourier_elim)
```

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

```
>$&fourier_elim([x^2-3*x>=0], [x])
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```
  Floating point error!
  Error in sqrt
  Try "trace errors" to inspect local variables after errors.
  g:
      useglobal; return sqrt(x^2-5x)/(2x-3)
  Error in:
  g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
      ^
```

Nb: Floating point error karena untuk x=1, g(x)=g(1) akan bernilai imajiner, yaitu

$$\frac{\sqrt{-2}}{2}$$

Silahkan Anda plot kurva fungsi di atas!

```
>function g(x):= sqrt(x^2-3*x)/(x+1);
>plot2d("g(x)",-10,10,-10,10):
>function g(x):= sqrt(x^2 - 4*x + 3)/(x - 2);
>plot2d("g(x)",-10,10,-10,10):
>function g(x):= sqrt(2*x^2 - 8*x + 6)/(x - 2);
>plot2d("g(x)", -1, 5, -10, 10):
```

Latihan Soal

$$\frac{\sqrt{3x^z - 12x + 9}}{x - 2}$$

Silakan plot kurva fungsi di atas untuk semua x dalam interval [-1,5]

```
>function g(x):= sqrt(3*x^2 - 12*x + 9)/(x - 2);
>plot2d("g(x)", -1, 5, -10, 10):
```

Fungsi Trigonometri

Fungsi trigonometri adalah suatu fungsi yang grafiknya berulang secara terus menerus dalam periode tertentu. Fungsi trigonometri mempelajari hubungan antara sudut-sudut dan sisi-sisi dalam segitiga.

Ada enam fungsi trigonometri dasar yang digunakan dalam Trigonometri. Enam fungsi trigonometri dasar adalah fungsi sinus , fungsi kosinus , fungsi sekan, fungsi co-sekan, fungsi tangen , dan fungsi cotangen . Sisi-sisi segitiga siku-siku adalah sisi tegak lurus, sisi miring, dan alas, yang digunakan untuk menghitung nilai sinus, cosinus, tangen, sekan , cosekan , dan kotangen menggunakan rumus trigonometri.

Rumus Fungsi Trigonometri

Rumus-rumus dasar untuk mencari fungsi-fungsi trigonometri adalah sebagai berikut:

$$sinx = \frac{depan}{miring}$$

$$cosx = \frac{samping}{miring}$$

$$tanx = \frac{depan}{samping}$$

$$cscx = \frac{miring}{depan}$$

$$secx = \frac{miring}{samping}$$

$$cotx = \frac{samping}{depan}$$

Seperti yang dapat kita amati dari rumus-rumus yang diberikan di atas, sinus dan cosekan merupakan kebalikan satu sama lain. Demikian pula, pasangan kebalikannya adalah cosinus dan sekan, dan tangen dan cotangen.

Fungsi Trigonometri dalam Empat Kuadran

Fungsi-fungsi trigonometri ini memiliki tanda numerik yang berbeda (+ atau -) di kuadran yang berbeda, yang didasarkan pada sumbu positif atau negatif kuadran. Fungsi trigonometri Sin dan Cosec positif di kuadran I dan II, dan negatif di kuadran III dan IV. Fungsi trigonometri Tan dan Cot positif hanya di Kuadran I dan III, dan rasio trigonometri Cos dan Sec positif hanya di kuadran I dan IV.

Grafik Fungsi Trigonometri

```
>plot2d("sin(x)",-2*pi,2*pi,grid=1);
>title("Plot_Sin(x)");
>xlabel("Sumbu x");
>ylabel("Sumbu y"):
>plot2d("cos(x)",-2*pi,2*pi,grid=1);
>title("Plot_Cos(x)");
>xlabel("Sumbu x");
>ylabel("Sumbu y"):
>plot2d("tan(x)",-2pi,2pi,grid=1);
>title("Plot_Tan(x)");
>xlabel("Sumbu x");
>ylabel("Sumbu y"):
```

Identitas Fungsi Trigonometri

- Identitas Timbal Balik

$$cosec(x) = \frac{1}{sin(x)}$$

$$sec(x) = \frac{1}{cos(x)}$$

$$cot(x) = frac1tan(x)$$

- Identitas Pythagoras

$$sin^2(x) + cos^2(x) = 1$$

$$1 + tan^2(x) = sec^2(x)$$

$$1 + cot^2(x) = cosec^2(x)$$

-Identitas Jumlah dan Selisih

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

$$cos(x + y) = cos(x)cos(y) - sin(x)sin(y)$$

$$tan(x + y) = \frac{tanx + tany}{1 - tanxtany}$$

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

$$cos(x - y) = cos(x)cos(y) + sin(x)sin(y)$$

$$tan(x-y) = \frac{tanx - tany}{1 + tanxtany}$$

- Identitas Tiga Sudut

$$Sin3x = 3sin(x) - 4sin(3x)$$

$$Cos3x = 4cos(3x) - 3cos(x)$$

$$Tan3x = \frac{3tan(x) - tan(3x)}{1 - 3tan(2x)}$$

- Identitas Produk

$$2sinxcosy = sin(x+y) + sin(x-y)$$

$$2cosxcosy = cos(x+y) + cos(x-y)$$

$$2sinxsiny = cos(x-y) - cos(x+y)$$

- Jumlah Identitas

$$sin(x) + sin(y) = 2sin(\frac{x+y}{2})cos(\frac{x-y}{2})$$

$$sin(x) - sin(y) = 2cos(\frac{x+y}{2})sin(\frac{x-y}{2})$$

$$cos(x) + cos(y) = 2cos(\frac{x+y}{2})cos(\frac{x-y}{2})$$

$$cos(x) - cos(y) = -2sin(\frac{x+y}{2}sin(\frac{x-y}{2})$$

Contoh Soal

```
>function f(x) := -2 cos 3(x)
>plot2d("-2 *cos (3*x)",-2*pi,2*pi,grid=1):
>function f(x) := -3 sin 2(x)
>plot2d("-3 *sin (2*x)",-2*pi,2*pi,grid=1):
```

Latihan Soal

1. Buatlah grafik f(x)= -4 sin 3(x)
2. f(x)=2 sin x, tentukan nilai f(0), f(1/2pi), f(4/3pi)

```
>function f(x) := -4 sin 3(x)
>plot2d("-4 *sin (3*x)",-2*pi,2*pi,grid=1):
>function f(x) := 2*sin(x)
>f(0), f(1/2pi), f(4/3pi)
```

```
  0
  0.316967773183
  0.823572192941
```

Fungsi Eksponensial

Definisi Fungsi Eksponen

Secara umum, fungsi eksponen didefinisikan sebagai:
Untuk a>0 dan x bilangan rill sembarang

$$a^x = e^{x.lna}$$

dimana a adalah bilangan basis atau pokok
Untuk meyakinkan definisi tersebut, kita akan menggunakan

$$3^2 = e^{2.ln3}$$

```
>3^2, exp(2*ln(3))
```

```
  9
  9
```

Latihan

$$2^5 = e^{5.ln2}$$

```
>2^5, exp(5*ln(2))
```

```
  32
  32
```

Sifat-Sifat Fungsi Eksponen

Apabila a>0, b>0, x dan y adalah bilangan rasional, maka

$$a^x a^y = a^{x+y};$$

$$\frac{a^x}{a^y} = a^{x-y};$$

$$(a^x)^y = a^{xy};$$

$$(ab)^x = a^x b^x;$$

$$\left(\frac{a}{b}\right)^x = \frac{a^x}{b^x}.$$

Bukti:

$$(a^x)^y = e^{y.lna^x} = e^{yx.lna} = a^{yx} = a^{xy}$$

```
>a:=2
```

```
  2
```

```
>x:=3
```

```
  3
```

```
>y:=4
```

```
  4
```

```
>ri:=(a^x)^y // ruas paling kiri
```

```
  4096
```

```
>ra:=a^(x*y) // ruas paling kanan
```

```
  4096
```

```
>exp1:=exp(y*ln(a^x))
```

```
  4096
```

```
>exp2:=exp(y*x*ln(a))
```

```
  4096
```

```
>exp3:=exp(x*y*ln(a))
```

```
  4096
```

```
>ri, ra, exp1, exp2, exp3
```

```
  4096
  4096
  4096
  4096
  4096
```

Latihan

$$(3^2)^3$$

```
>a:=3
```

```
  3
```

```
>x:=2
```

```
  2
```

```
>y:=3
```

```
  3
```

```
>ri:=(a^x)^y
```

729

```
>ra:=a^(x*y)
```

729

```
>exp1:=exp(y*ln(a^x))
```

729

```
>exp2:=exp(y*x*ln(a))
```

729

```
>exp3:=exp(x*y*ln(a))
```

729

```
>ri, ra, exp1, exp2, exp3
```

729
729
729
729
729

Grafik Fungsi Eksponen

Bentuk umum dari grafik fungsi eksponen:

$$y = f(x) = ka^x$$

dimana k adalah konstanta dan a adalah basis atau bilangan pokok.
Syarat dari basis adalah

$$a > 0$$

$$a \neq 1$$

$$0 < a < 1$$

$$a > 1$$

Grafik fungsi eksponen akan selalu simteri terhadap sumbu-y. Selain itu, grafik fungsi eksponen ini selalu memotong sumbu-y di titik (0,k).
Grafik fungsi eksponen ini memiliki 2 jenis:
1. Grafik fungsi Monoton Naik

$$a > 1$$

2. Grafik fungsi Monoton Turun

$$0 < a < 1$$

Contoh Grafik Monoton Naik

$$y = 2^x$$

```
>x=[-3,-2,-1,0,1,2,3]; y=2^x; plot2d(x,y,"0-"); xlabel("X"); ...
>ylabel("Y"); title("Grafik Eksponensial y=2^x"); ...
>grid:=true:
```

Contoh Grafik Monoton Turun

$$y = (\frac{1}{2})^x$$

```
>x=[-3,-2,-1,0,1,2,3]; y=(1/2)^x; plot2d(x,y,"0-",color(green)); xlabel("X"); ...
>ylabel("Y"); title("Grafik Eksponensial y=(1/2)^x"); ...
>grid:=true:
```

Latihan

$$3^{-x}+1$$

x=[-4,-2,0,2,4]

```
>x=[-4,-2,0,2,4]; y=3^(-x)+1; plot2d(x,y,"0-",color(green)); xlabel("X"); ...
>ylabel("Y"); title("Grafik Eksponensial y=3^(-x)+1"); ...
>grid:=true:
```

Fungsi Logaritma

Definisi Logaritma

$$^{a}log(x)=y$$

atau

$$a^{y}=x$$

dengan

$$a>0$$

$$a\neq 1$$

$$x>0$$

serta a disebut basis dan x disebut numerus.

Definisi Fungsi Logaritma

$$f(x) = ^a log(x)$$

dengan

$$a > 0$$

$$a \neq 1$$

$$x > 0$$

serta a disebut basis dan x disebut numerus.

Sifat-Sifat Logaritma

$$^a log(xy) = ^a logx + ^a logy$$

$$^a log\frac{x}{y} = ^a logx - ^a logy$$

$$^a logx^n = n.^a logx$$

$$a^{^a logx} = x$$

Jika a>1 dan

$$^a logx > ^a logy, maka, x > y$$

Jika 0<1 dan

$$^a logx < ^a logy, maka, x > y$$

Sifat Grafik Fungsi Logaritma

$$D_f = (0, \infty)$$

$$R_f = \mathbb{R}$$

Tidak memotong sumbu y
Memotong sumbu x di titik (1,0)
Jika 0<a<1 maka grafik fungsi turun dan

$$x \to 0^+$$

maka

$$y \to \infty$$

Jika a>1 maka grafik fungsi naik dan

$$x \to 0^+$$

maka

$$y \to -\infty$$

Latihan Membuat Grafik Logaritma

```
>function k(x) := logbase(x, 7)
>k(49)
```

```
  2
```

```
>plot2d("k(x)"):
```

Komposisi Fungsi

Komposisi fungsi adalah susunan beberapa fungsi yang berkaitan dan terhubung. Bahasa paling mudah untuk mendeskripsikan komposisi fungsi ini adalah "fungsi di dalam fungsi". Sebagai contoh, jika

$$f(x) = x^3 + 1$$

dan

$$g(x) = x^2 + x$$

maka

$$(f \circ g)(x) = f(g(x)) = (\boxed{x^2 + x})^3 + 1$$

$$(g \circ f)(x) = g(f(x)) = (\boxed{x^3 + 1})^2 + \boxed{x^3 + 1}$$

```
>function f(x):= x^3+1
>function g(x):= x^2+x;
>f(g(5)) // komposisi fungsi
```

```
  27001
```

```
>g(f(5))
```

```
  16002
```

```
>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))
```

```
  27001
```

```
>function u(x) := g(f(x))
```

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>plot2d(["h(x)", "u(x)"], -10,10,0,10):
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
  [1,  2,  9,  28,  65,  126,  217,  344,  513,  730,  1001]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
  [1,  2,  9,  28,  65,  126,  217,  344,  513,  730,  1001]
```

```
>gmap(200:210)
```

```
  [40200,  40602,  41006,  41412,  41820,  42230,  42642,  43056,  43472,
  43890,  44310]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

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

```
  endfunction
```

```
>f(1)
>f(-2)
>f(-5:5)
>aspect(1.5); plot2d("f(x)",-5,5):
```

```
  A function returned no value. Cannot assign this to a variable.
   %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());
```

```
>function f(x) &= 2*E^x // fungsi simbolik
```

```
   x
2 E
```

```
>$f(a) // nilai fungsi secara simbolik
>f(E) // nilai fungsi berupa bilangan desimal
```

```
  30.308524483
```

```
>$f(E), $float(%)
>function g(x) &= 3*x+1
```

```
3 x + 1
```

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

```
     3 x + 1
2 E
```

```
>plot2d("h(x)",-1,1):
```

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik tersebut.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

1. Diberikan sebuah fungsi

$$a(x) = x^2 + x + 11$$

tentukan nilai
a. a(-2)
b. a(-12)

```
>function a(x) := x^2+x+11
>a(-2), a(-12)
```

```
  13
  143
```

```
>plot2d("a",-10,10):
```

2. Diberikan sebuah fungsi

$$b(x) = \frac{x^2 + 36}{x + 6}$$

hitunglah masing-masing nilai b(4) dan b(44)

```
>function b(x) := (x^2+36)/(x-6)
>b(4), b(44)
```

```
  -26
  51.8947368421
```

```
>plot2d("b",-100,100,-100,100):
```

3. Diberikan sebuah fungsi

$$p(x) = x^3 - 12x^2 + 48x - 64$$

tentukan nilai p(4), p(-4), p(10)

```
>function p(x) := x^3-12*x^2+48*x-64
>p(4), p(-4), p(10)
```

```
  0
  -512
  216
```

```
>plot2d("p",-1,8):
```

4. Tentukan nilai f(41) dari fungsi berikut

$$f(x) = \sqrt{x^2 - 81}$$

```
>function f(x) := sqrt(x^2-81)
>f(41)
```

```
  40
```

```
>plot2d("f",-50,50):
```

5. Untuk fungsi

$$f(x) = x^2 - 5x + 6 \text{ dan } g(x) = x + 7$$

tentukan nilai (f o g)(0) dan (g o f)(10)

```
>function f(x) := x^2-5*x+6; $f(x)
>function g(x) := x+7; $g(x)
>f(g(-2)), g(f(0))
```

```
  6
  13
```

```
>plot2d("f(g(x))",-5,5):
>plot2d("g(f(x))",-5,5):
```

Limit

Untuk sebuah fungsi f(x), dikatakan bahwa limit
(x) mendekati L saat x mendekati, ditulis sebagai:

$$\lim_{x \to c} f(x) = L$$

$$\forall \epsilon > 0, \exists \delta > 0 \text{ sehingga jika } 0 < |x - c| < \delta, \text{ maka } |f(x) - L| < \epsilon.$$

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf).

Perhatikan!!!

```
>$limit(sqrt(x^2-3*x)/(x+1),x,inf)
>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
>$showev('limit((3*x+5)/(2*x-4),x,minf))
>$showev('limit(1/(2*x-1),x,0))
```

Latihan soal

Contoh soal, hitung nilai limitnya

$$lim_{x \to 0} \frac{1}{x^2 + 1}$$

```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x\to\infty} \frac{4x+7}{x-1}$$

```
>$showev('limit((4*x+7)/(x-1),x,inf))
```

limit kiri dan kanan

Limit Kiri :

$$\lim_{x\to c^-} f(x)$$

menggunakan opsi "minus"
Limit Kanan :

$$\lim_{x\to c^+} f(x)$$

menggunakan opsi "plus"
Jika,

$$\lim_{x\to c^-} f(x) \neq \lim_{x\to c^+} f(x)$$

maka,

$$\lim_{x\to c} f(x) tidakada$$

contoh:

$$\lim_{x\to 1} \frac{|x-1|}{x-1}$$

```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
>$showev('limit(abs(x-1)/(x-1),x,1,plus))
```

karena limit dari kiri dan limit dari kirinya berbeda maka nilai limit di x=1 tidak ada

Latihan soal

1.

$$\lim_{x\to 0} \frac{1}{x}$$

2.

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

hitunglah limit kanan dan kirinya dan berikan kesimpulan

```
>$showev('limit(1/(x),x,0,minus))
>$showev('limit(1/(x),x,0,plus))
>$showev('limit((x^2-4)/(x-4),x,2,minus))
>$showev('limit((x^2-4)/(x-4),x,2,plus))
>plot2d("(x^2-4)/(x-4)"):
```

Limit

limit dalam fungsi aljabar

$$\lim_{x \to 5} 2x - 8$$

hitunglah limit dari kanan, limit dari kiri, dan x mendekati 5
menggunakan perintah $showev dan jika dari kiri menggunakan opsi "minus" dan jika dari kanan menggunakan opsi "plus"

```
>$showev('limit((2*x-8),x,5,minus))
>$showev('limit((2*x-8),x,5,plus))
>$showev('limit((2*x-8),x,5))
>$showev('limit((x^2/x),x,0,plus))
>$showev('limit((x^2/x),x,0,minus))
```

Limit dalam berbagai fungsi

Limit dalam fungsi aljabar

$$\lim_{x \to 4} \frac{\sqrt{x} - 2}{x - 4}$$

hitunglah nilai limit dari fungsi aljabar tersebut

```
>$showev('limit((sqrt(x)-2)/(x-4),x,4))
```

Latihan:
hitunglah nilai dari limit fungsi berikut

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

```
>$showev('limit((sqrt(1-2*x))/((3*x+2)^3),x,-2))
>plot2d("(sqrt(1-2*x))/((3*x+2)^3)"):
```

Limit dalam fungsi trigonometri

$$\lim_{x\to 0} \frac{\sin(2x)}{x} = 2$$

hitunglah nilai limit dari fungsi trigonometri tersebut

```
>$showev('limit((sin(2*x))/(x),x,0))
```

Latihan;
hitunglah nilai limit dari fungsi tersebut

$$\lim_{x\to \frac{\pi}{2}} \sin(x)$$

$$\lim_{x\to 0} \frac{1-\cos(5x)}{x^2}$$

```
>$showev('limit((sin(x)),x,(pi/2)))
>plot2d("(sin(x))"):
>$showev('limit(((1-(cos(5*x)))/(x^2)),x,0))
>plot2d("(1-(cos(5*x)))/(x^2)"):
```

Limit fungsi trigonometri

$$\lim_{x\to \frac{\pi}{2}} \frac{\log(x)}{x^2}$$

hitunglah nilai limit trigonometri tersebut

```
>$showev('limit((log(x)/x^2),x,(pi/2)))
```

Latihan:
Hitunglah nilai limit dari fungsi berikut
baris tak tentu

$$\lim_{x\to 1} \frac{\log(x)}{x-1}$$

```
>$showev('limit(log(x)/(x-1),x,1))
>plot2d("(log(x)/(x-1))"):
```

Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
>p &= expand((x+h)^2-x^2)|simplify; $p //pembilang dijabarkan dan disederhanakan
>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
>$limit(q,h,0) // nilai limit sebagai turunan
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

Petunjuk: ekspansikan (x+h)^n dengan menggunakan teorema binomial.

Akan ditunjukkan bahwa $f'(x) = \lim_{h \to 0} \frac{(x+h)^n - x^n}{h} = nx^{n-1}$

Pertama, ekspansikan $(x+h)^n$, yakni:

$$(x+h)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} h^k$$

$$\Leftrightarrow (x+h)^n = \binom{n}{0} x^n + \binom{n}{1} x^{n-1} h + \binom{n}{2} x^{n-2} h^2 + ... + \binom{n}{n} h^n$$

$$\Leftrightarrow (x+h)^n = x^n + nx^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \binom{n}{3} x^{n-3} h^3 + ... + h^n$$

Sehingga, $f'(x)$ menjadi: $f'(x) = \lim_{h \to 0} \frac{(x+h)^n - x^n}{h}$

$$\Leftrightarrow f'(x) = \lim_{h \to 0} \frac{x^n + nx^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \binom{n}{3} x^{n-3} h^3 + ... + h^n - x^n}{h}$$

$$\Leftrightarrow f'(x) = \lim_{h \to 0} nx^{n-1} + \binom{n}{2} x^{n-2} h + \binom{n}{3} x^{n-3} h^2 + ... + h^{n-1}$$

$\Leftrightarrow f'(x) = nx^{n-1}$. Terbukti.

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

Petunjuk: ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.

Akan ditunjukkan bahwa $\lim_{h\to 0} \frac{sin(x+h) - sin(x)}{h} = cos(x)$

Pertama, gunakan rumus jumlah sudut untuk $sin(x+h)$, yakni:

$$sin(x+h) = sin(x)cos(h) + cos(x)sin(h)$$

$$\Longleftrightarrow \lim_{h\to 0} \frac{\sin(x+h) - \sin(x)}{h} = \lim_{h\to 0} \frac{[\sin(x)\cos(h) + \cos(x)\sin(h)] - \sin(x)}{h}$$

$$\Longleftrightarrow \lim_{h\to 0} \frac{\sin(x)\cos(h) + \cos(x)\sin(h) - \sin(x)}{h}$$

$$\Longleftrightarrow \lim_{h\to 0} \left(\frac{\sin(x)[\cos(h) - 1]}{h} + \frac{\cos(x)\sin(h)}{h} \right)$$

Diketahui bahwa:

$$1).\ \lim_{h\to 0} \frac{cos(h) - 1}{h} = 0$$

$$2).\ \lim_{h\to 0} \frac{\sin(h)}{h} = 1$$

sehingga:

$$\Longleftrightarrow \lim_{h\to 0} [\sin(x)(0) + \cos(x)(1)]$$

$\Leftrightarrow 0 + cos(x) = cos(x)$. Terbukti.

Latihan

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

```
>function f(x) := x^2
>$showev('limit((((x+h)^2-x^2)/h),h,0)) // turunan x^2
>function df(x) &= limit((((x+h)^2-x^2)/h),h,0);  $df(x)// df(x) = f'(x)
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",2,0.
```

2.

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

```
>function f(x) := x^2+3*x
>$showev('limit(((x+h)^2+3*(x+h)-(x^2+3*x))/h,h,0)) // turunan x^2+3*x
>function df(x) &= limit(((x+h)^2+3*(x+h)-(x^2+3*x))/h,h,0);  $df(x)// df(x) = f'(x)
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.4), label("df(x)",1,-0
>function f(x) := log(x)
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

Akan ditunjukkan bahwa $\lim_{h\to 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$

$$\lim_{h\to 0} \frac{\log(x+h) - \log(x)}{h}$$

Diketahui bahwa:

$$\log(a) - \log(b) = \log(\frac{a}{b})$$

Sehingga:

$$= \lim_{h\to 0} \frac{\log\left(\frac{x+h}{x}\right)}{h}$$

$$= \lim_{h\to 0} \frac{\log\left(1+\frac{h}{x}\right)}{h}$$

Diketahui bahwa:

$$\lim_{h\to 0} \frac{\log(1+u)}{u} = 1$$

$$\text{dengan } u = \frac{h}{x}$$

Sehingga:

$$f'(x) = \frac{1}{x} \times x$$

$$f'(x) = \frac{1}{x}$$

$\lim_{h\to 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$. Terbukti.

```
>function df(x) &= limit(((log(x+h)-log(x))/h) ,h,0);  $df(x)// df(x) = f'(x)
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",2,0.
```

Turunan Eksponensial

$$\lim_{h\to 0} \frac{e^{x+h} - e^x}{h}.$$

```
>function f(x) := E^x
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

Menggunakan factor atau pemfactoran pada

$$e^x$$

dikarenakan untuk menyederhanakan limit, sehingga kita bisa memisahkan komponen yang bergantung pada h dan yang tidak bergantung pada h.

$$\text{Akan ditunjukkan bahwa } \lim_{h\to 0}\frac{e^{x+h}-e^{x}}{h}=1$$

$$f'(x)=\lim_{h\to 0}\frac{e^{x+h}-e^{x}}{h}=1$$

$$f'(x)=\lim_{h\to 0}\frac{e^{x}\cdot e^{h}-e^{x}}{h}$$

$$f'(x)=\lim_{h\to 0}\frac{e^{x}(e^{h}-1)}{h}$$

$$f'(x)=e^{x}\lim_{h\to 0}\frac{e^{h}-1}{h}$$

karena turunan dari

$$e^{x}, x=0$$

adalah

$$e^{0}=1$$

maka,

$$\lim_{h\to 0}\frac{e^{h}-1}{h}=1$$

Sehingga:

$$f'(x)=e^{x}$$

```
>function df(x) &= E^x
```

```
  x
E
```

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.3), label("df(x)",1,-0
>function f(x) &= x^x
```

$$x^x$$

```
>$showev('limit(f(x),x,0))
>plot2d("x^x"):
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

Dapat dilihat bahwa dari yang kita cari pertama nilai limit dari

$$x^x$$

itu ada yaitu 1.

Sementara pada yang kita cari menggunakan definisi turunan nilai limitnya infinity.

Begitu juga pada grafik, nilai dari x hanya terdefinisi di zona positif, jadi untuk mencari turunannya perlu diasumsikan dulu bahwa nilai x nya itu berjalan dengan x yang positif karena nilai dari yang negatifnya tidak terdefinisi.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

Akan ditunjukkan bahwa $\lim_{h\to 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x(\log(x)+1)$

Menggunakan logaritma untuk menyederhanakan

$$y = x^x$$

$$log(y) = log(x^x) = xlog(x)$$

$$\frac{1}{y}\frac{d}{dx} = \frac{d}{dx}(xlog(x))$$

$$\frac{1}{y}\frac{d}{dx} = \log(x) + 1$$

Substitusi kembali

$$y = x^x, sehingga$$

$$\frac{d}{dx} = x^x(log(x)+1)$$

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

```
[x > 0]
```

```
>&forget(x<0)
```

```
[x < 0]
```

```
>&facts()
```

```
[kind(sinh, one_to_one), kind(log, one_to_one),
                kind(tanh, one_to_one), kind(log, increasing)]
```

Latihan

$$e^{3x} + 4xe^x$$

```
>function f(x) &= E^(3*x)+4*x*E^x
```

```
   3 x        x
  E    + 4 x E
```

```
>$showev('limit(factor((E^(3*x+h)+4*(x+h)*E^(x+h)-E^(3*x)-4*x*E^x)/h),h,0))
>function df(x) &= E^x*(E^(2*x)+4*x+4)
```

```
    x   2 x
  E  (E    + 4 x + 4)
```

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",1,-0
```

2.

$$log(3x+1)$$

```
>function f(x) &= log(3*x+1)
```

$$\log(3\ x\ +\ 1)$$

```
>$showev('limit((log(3*x+h+1)-log(3*x+1))/h,h,0))
>function df(x) &= limit(((log(3*x+h+1)-log(3*x+1))/h) ,h,0);  $df(x)// df(x) = f'(x)
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",1,-0
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

$$f'(x) = \lim_{h \to 0} \frac{arcsin(x+h) - arcsinx}{h}$$

$$asumsikanarcsin(x+h) = Adanarcsinx = B$$

$$sinA = x + h, sinB = x$$

$$sinA - sinB = (x+h) - x$$

$$\frac{d}{dx}(arcsinx) = \lim_{A \to B} \frac{A-B}{SinA - SinB}$$

$$\frac{d}{dx}(arcsinx) = \lim_{A \to B} \frac{A-B}{2Cos\frac{A+B}{2} - 2Sin\frac{A-B}{2}}$$

$$\frac{d}{dx}(arcsinx) = \lim_{A \to B} \frac{\frac{A-B}{2}}{Sin\frac{A-B}{2}} \times \frac{1}{Cos\frac{A+B}{2}}$$

$$\frac{d}{dx}(arcsinx) = 1 \times \frac{1}{Cos\frac{B+B}{2}}$$

$$\frac{d}{dx}(arcsinx) = \frac{1}{CosB}$$

$$sin2y + cos2y = 1$$

$$cosB = \frac{1}{\sqrt{1 - sin^2 B}} = \frac{1}{\sqrt{1 - x^2}}$$

$$f'(x) = \frac{dy}{dx} = \frac{1}{\sqrt{1 - x^2}}$$

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

$$f'(x) = \lim_{h \to 0} \frac{tan(x+h) - tan(x)}{h}$$

$$tan(a+b) = \frac{tan(a) + tan(b}{1 - tan(a)tan(b)}$$

$$tan(x+h) = \frac{tan(x) + tan(h}{1 - tan(x)tan(h)}$$

$$tan(x+h) - tan(x) = \frac{tan(x) + tan(h}{1 - tan(x)tan(h)} - tan(x)$$

$$\frac{tan(x) + tan(h) - tan(x)(1 - tan(x)tan(h))}{1 - tan(x)tan(h)} = \frac{tan(h) + tan^2(x)tan(h)}{1 - tan(x)tan(h)}$$

$$= \frac{tan(h)(1 + tan^2(x))}{1 - tan(x)tan(h)}$$

$$= \lim_{h \to 0} \frac{h(1 + tan^2(x))}{1 - tan(x)h}$$

$$= \lim_{h \to 0} \frac{h(1 + tan^2(x))}{1}$$

$$= \lim_{h \to 0} h(1 + tan^2(x))$$

$$= \lim_{h \to 0} \frac{h(1 + tan^2(x))}{h}$$

$$= 1 + tan^2(x)$$

$$= 1 + tan^2(x) = sec^2(x)$$

$$f'(x) = sec^2(x) = \frac{1}{cos^2(x)}$$

Jadi, terbukti benar bahwa

$$= f'(x) = \lim_{h \to 0} \frac{tan(x+h) - tan(x)}{h} = \frac{1}{cos^2(x)}$$

$$\frac{d}{dx}(arcsinx) = \lim_{A \to B} \frac{A - B}{2Cos\frac{A+B}{2} - 2Sin\frac{A-B}{2}}$$

$$\frac{d}{dx}(arcsinx) = \lim_{A \to B} \frac{\frac{A-B}{2}}{Sin\frac{A-B}{2}} \times \frac{1}{Cos\frac{A+B}{2}}$$

$$= \frac{d}{dx}(arcsinx) = 1 \times \frac{1}{Cos\frac{B+B}{2}}$$

$$= \frac{d}{dx}(arcsinx) = \frac{1}{CosB}$$

$$sin2y + cos2y = 1$$

$$cosB = sqrt1\breve{}sin^2B = \sqrt{1} - x^2$$

$$= f'(x) = \frac{dy}{dx} = \frac{1}{\sqrt{1-x^2}}$$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

```
sinh(x)
```

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

Hasilnya adalah cosh(x), karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
>function f(x) &= sin(3*x^5+7)^2
```

```
           2     5
       sin (3 x  + 7)
```

```
>diff(f,3), diffc(f,3)
```

```
  1198.32948904
  1198.72863721
```

Apakah perbedaan diff dan diffc?

diff untuk menghitung turunan numerik ke-n dari fungsi f menggunakan metode perbedaan terbatas (finite difference). Sedangkan diffc mungkin merujuk pada metode yang lebih canggih atau terkompensasi untuk menghitung turunan numerik ke-n dari fungsi f.

```
>$showev('diff(f(x),x))
>$% with x=3
>$float(%)
>plot2d(f,0,3.1):
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendifinisikan fungsi f
```

$$5 \cos(2\ x) - 2\ x \sin(2\ x)$$

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

$$-12 \sin(2\ x) - 4\ x \cos(2\ x)$$

```
>$'f(1)=f(1), $float(f(1)), $'f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

```
  1.35822987384
```

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

```
  0
  -5.67530133759
```

```
>plot2d(["f(x)","df(x)"],0,2*pi,color=[blue,red]): //grafik fungsi dan turunannya
```

Perhatikan titik-titik "puncak" grafik y=f(x) dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

Integral

Integral Tak Tentu

DEFINISI

Integral Tak Tentu (undefinite integral) adalah bentuk integral yang

tidak memiliki batas tertentu, sehingga hasilnya adalah fungsi umum yang disertai konstanta integrasi (C). Integral tak tentu merupakan kebalikan dari turunan, sehingga disebut antiturunan atau antiderivatif.

Integral tak tentu dari suatu fungsi f(x) adalah fungsi F(x) yang

mempunyai turunan f(x), maka integral tak tentu merupakan himpunan anti turunan F(x) dari f(x) pada interval negatif tak hingga sampai tak hingga yang dinotasikan :

$$f(x) = \int F(x)dx + C$$

dimana F'(x)=f(x) dan C adalah konstanta integrasi

KURVA FUNGSI ANTIDERIVATIF

Kurva fungsi antiderivatif adalah kurva yang menggambarkan semua

fungsi yang memiliki turunan sama. Setiap fungsi antiderivatif dari suatu fngsi f(x) berbeda hanya dalam konstanta integrasi (C)

1. kurva antiderivatif dari fungsi aljabar dengan

$$f(x) = 3x^2$$

```
>$showev('integrate(3*x^2,x)+c)
>plot2d(["3*x^2","x^3","x^3+1","x^3+2","x^3+3"]):
```

2. kurva antiderivatif dari fungsi trigonometri dengan

$$f(x) = sinx$$

```
>$showev('integrate(sin(x),x)+c)
>plot2d(["-cos(x)","-cos(x)+1","-cos(x)+2","-cos(x)+3","-cos(x)+4"]):
```

3. kurva antiderivatif dari fungsi eksponensial dengan

$$f(x) = xe^x$$

```
>$showev('integrate(x*E^x,x)+c)
>plot2d(["x*E^x","(x-1)E^x+1","(x-1)E^x+2","(x-1)E^x+3","(x-1)E^x+4"]):
```

4. kurva antiderivatif dari fungsi logaritma dengan

$$f(x) = log(8x)$$

```
>$showev('integrate(log(8*x),x)+c)
>plot2d(["log(8*x)","(8*x)log(8*x)-8*x/8 +1","(8*x)log(8*x)-8*x/8 +2","(8*x)log(8*x)-8*x/8
```

5. kurva antiderivatif dari fungsi komposisi dengan

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

$$g(x) = 2x - 1$$

dengan

$$f(g(x))$$

```
>function f(x) &&= x+2
```

```
x + 2
```

```
>function g(x) &&= 2*x-1
```

```
2 x - 1
```

```
>$showev('integrate(f(g(x)),x)+c)
>plot2d(["2*x+1","x^2+x+1","x^2+x+2","x^2+x+3","x^2+x+4"]):
```

SIFAT-SIFAT INTEGRAL TAK TENTU

Integral tak tentu adalah LINEAR. Perlu diingat pada Aturan Kelipatan Konstanta, bahwa

$$D_x$$

adalah suatu operator linear. Ini berarti dua hal.

$$D_x[k.f(x)] = k.D_x f(x)$$

$$D_x[f(x) + g(x)] = D_x f(x) + D_x g(x)$$

Dari dua sifat ini, sifat ketiga mengikuti secara otomatis.

$$D_x[f(x) - g(x)] = D_x f(x) - D_x g(x)$$

Apa yang benar untuk anti turunan adalah benar juga untuk integral tak tentu (anti-turunan).

(Kelinearan dari

$$\int ...dx).$$

Andaikan f dan k mempunyai anti turunan (integral tak tentu) dan andaikan k suatu konstanta, maka:

$$\int k.f(x)dx = k \int f(x)dx;$$

$$\int [f(x) + g(x)]dx = \int f(x)dx + \int g(x)dx;$$

$$\int [f(x) - g(x)]dx = \int f(x)dx - \int g(x)dx$$

>

Bukti: Untuk memperlihatkan sifat (i) dan (ii), kita cukup mendiferensikan ruas kanan dan amati bahwa kita akan memperolah integran dari ruas kiri.

$$D_x[k. \int f(x)dx] = k.D_x \int f(x)dx = k.f(x)$$

$$D_x[\int f(x)dx + \int g(x)dx] = D_x \int f(x)dx + D_x \int g(x)dx = f(x) + g(x)$$

Contoh Soal

$$\int (3x^2 + 4x)dx$$

```
>$showev('integrate(3*x^2,x)+c)
>$showev('integrate(4*x,x)+c)
>$showev('integrate(3*x^2+4*x,x)+c)
```

Apabila perhitungan secara manual:
image: contoh1.jpg

Latihan

$$\int (3sint - 2cost)dt$$

```
>$showev('integrate(3*sin(t),t)+c)
>$showev('integrate(-2*cos(t),t)+c)
>$showev('integrate(3*sin(t)-2*cos(t),t)+c)
```

RUMUS-RUMUS INTEGRAL TAK TENTU

$$\int a.x^n dx = \frac{a}{n+1}x^{n+1} + C$$

dimana a adalah konstanta dan C juga merupakan konstanta.

$$\int \frac{a}{x}dx = \int ax^{-1}dx = aln|x| + C$$

dimana a adalah konstanta dan C juga merupakan konstanta.

Contoh Soal

$$\int \frac{1}{x^2+1} dx$$

```
>$showev('integrate(1/(1+x^2),x)+c)
```

Mengapa arctan(x)+c?
Jadi, antara

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

dengan arctan(x) ini saling berhubungan dengan definisi invers trigonometri.
Fungsi arctan(x) adalah fungsi invers dari tan(x). Secara matematis, turunan dari arctan(x) adalah:

$$\frac{d}{dx} arctan(x) = \frac{1}{x^2+1}$$

Ini berarti bahwa arctan(x) adalah fungsi yang jika diturunkan menghasilkan

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

Oleh karena itu, ketika kita melakukan integral terhadap

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

kita mendapatkan kebalikan dari turunan tersebut, yaitu fungsi arctan(x).

Integral tentu

Integral tentu adalah integral yang memiliki batasan atas dan bawah yang jelas, sehingga menghasilkan sebuah nilai. Batasan dari integral tentu adalah a sampai b atau batas atas sampai batas bawah.
Integral tentu dapat dihitung dengan menggunakan Teorema Dasar Kalkulus

$$\int_a^b (f(x))dx = F(b) - F(a), \quad \text{dengan } F'(x) = F(x)$$

Misalnya mencari hasil dari fungsi integral:

$$\int_2^3 6x + 2 \, dx$$

dengan menggunakan rumus kita peroleh:

$$\int_2^3 6x + 2\ dx = [3x^2 + 2x]_2^3 = 3(3)^2 + 2(3) - 3(2)^2 + 2(2) = 33 - 16 = 17.$$

```
>$showev('integrate(6*x+2,x,2,3))
```

Latihan:

a.

$$\int_0^1 x^2 - 4x + 3\ dx$$

b.

$$\int_2^6 1 + cos(t)\ dt$$

Integral dapat diselesaikan dengan EMT yaitu dengan fungsi "integrate". EMT juga dapat digunakan untuk menghitung integral tentu dengan beberapa fungsi yang mengimplementasikan algoritma kuadratur(perhitungan integral tentu menggunakan metode numerik).

Fungsi "showev" digunakan untuk menampilkan atau melihat hasil perhitungan. Setelah fungsi integrate, diikuti fungsi, terhadap variabel apa, batas bawah, dan batas atas.

```
>$showev('integrate(x^2-4*x+3,x,0,1))
>$showev('integrate(1+cos(t),t,2,6))
```

- fx &=: Ini adalah operasi penugasan gabungan. Artinya, hasil dari perhitungan di sebelah kanan akan ditambahkan ke variabel fx. Jika fx belum didefinisikan sebelumnya, maka fx akan dibuat sebagai sebuah list.
-makelist(...,i,1,length(t)): Fungsi ini digunakan untuk membuat sebuah list.

Parameter-parameternya adalah:
- f(t[i]+0.1): Ini adalah ekspresi yang akan dihitung untuk setiap nilai i. Fungsi f akan dievaluasi pada nilai t[i] ditambah 0.1.
- i: Adalah indeks yang akan digunakan untuk mengakses elemen-elemen dalam list t.
- 1: Adalah nilai awal untuk indeks i.
- length(t): Adalah nilai akhir untuk indeks i, yaitu panjang dari list t.

```
>function map f(x) &= E^(-x^2)
```

```
          2
       - x
      E
```

```
>$showev('integrate(f(x),x))
```

Fungsi f diatas tidak memiliki antiturunan yang dapat diungkapkan dalam bentuk tertutup, sehingga kita tidak dapat menghitungnya dengan Teorema Dasar Kalkulus. KIta dapat menghitung integral tentu fungsi tersebut dengan menggunakan metode numerik(rumus kuadratur).

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```

Integral tentu erat kaitannya dengan luas daerah dibawah kurva, sehingga menghitung fungsi diatas dapat dihampiri dengan jumlah luas persegi panjang-persegi panjang di bawah kurva tersebut. Cara menghitungnya adalah dengan langkah-langkah sebagai berikut:
Langkah pertama, yaitu kita dapat menyimpan nilai-nilai x ke dalam variabel misalkan t.

```
>t &= makelist(a,a,0,pi-0.1,0.1);
```

Selanjutnya kita dapat menyimpan nilai dari masing-masing f(x).

```
>fx &= makelist(f(t[i]+0.1),i,1,length(t));
>0.1*sum(f(x+0.1))
```

```
  0.836219610253
```

jadi hasilnya adalah:

maxima: 'integrate(f(x),x,0,pi) = 0.1*sum(fx[i],i,1,length(fx))

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai f(x) untuk x = 0.1, 0.2, 0.3, ..., 3.2.

Jika kita memperbanyak jumlah persegi panjang dibawah kurva, maka lebar sub-intervalnya akan semakin kecil dan persegi panjangnya akan menutup daerah dibawah kurva dengan sempurna.Untuk mendapatkan hasil yang mendekati nilai sebenarnya kita dapat gunakan misalnya lebar sub-interval misalnya 0.01, atau jika ingin lebih presisi lagi bisa 0.001 dan seterusnya. Kita akan mencoba menghitung integral fungsi diatas dengan dengan t yang diperkecil lagi menjadi 0.01.

```
>x=0:0.01:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,pi,>add):
```

Gambar diatas jika dibandingkan dengan gambar sebelumnya, terlihat perbedannya dimana jika gambar pertama daerah dibawah kurva masih belum tertutup sempurna, tetapi pada gambar diatas terlihat bahwa daerah dibawah kurva lebih tertutup sempurna dan tidak terlihat daerah yang masih berlubang.
Jika kita hitung integral fungsinya maka akan diperoleh hasil yang mendekati sebenarnya.

```
>t &= makelist(a,a,0,pi-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>0.01*sum(f(x+0.01))
```

```
  0.881219234865
```

Integral fungsi diatas yang memiliki batas tidak dapat kita hitung langsung secara eksak, tetapi ternyata jika kita mengubah batasnya menjadi tak hingga kita dapat menghitungnya secara eksak. Berikut akan ditunjukkan contohnya.

```
>$showev('integrate(f(x),x,0,inf))
```

Contoh lain fungsi tidak memiliki antiderivatif yang hanya dapat dihitung dengan metode numerik adalah sebagai berikut.

```
>function f(x) &= x^(2*x)
```

$$x^{2\,x}$$

```
>$showev('integrate(f(x),x,0,2))
>x=0:0.002:2-0.002; plot2d(x,f(x+0.002),>bar); plot2d("f(x)",0,2,>add):
>k &= makelist(a,a,0,pi-0.002,0.002);
>fx &= makelist(f(k[i]+0.002),i,1,length(k));
>0.002*sum(f(x+0.002))
```

```
  5.54658986236
```

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin\left(3\,x^5+7\right)^2$$

```
>integrate(f,0,1)
```

```
  0.542581176074
```

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin\left(3\,x^5+7\right)^2$$

```
>integrate(f,0,1)
```

```
  0.542581176074
```

Berikut adalah contoh integral fungsi eksponensial yang dapat dihitung secara eksak.

```
>$showev('integrate(x*exp(-x),x,0,1))
```

Latihan:

tentukan integral dari
a.

$$3x^2 + 4x + 2, \quad \text{dari } x = 1 \quad \text{sampai } x = 2$$

b.

$$sin(x), \quad \text{dari } x = 0 \quad \text{sampai } x = pi$$

```
>$showev('integrate (3*x^2+4*x+2,x,1,2))
>$showev('integrate (sin(x),x,0,pi))
```

Aplikasi Integral Tentu

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add):
>b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x):
```

Menggunakan perintah diatas kita mendapatkan penyelesaian dari gabungan kedua fungsi diatas yang kemudian disimpan dalam variabel b. Kemudian dibuat baris array x yang berisi 200 nilai dengan interval dari 0 sampai b untuk kemudian dibali urutan nilai-nilainya menjdai baris baru yaitu xi.

```
>plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add):
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

```
  0
  0.61803398875
```

Hasil diatas merupakan absis dari kedua titik potong kurva.

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

```
  0.0758191713542
```

Hasil perhitungan tadi jika dibandingkan dengan hasil perhitungan analitik akan menghasilkan nilai perhitungan yang sama. Kita coba dengan perhitungan analitik sebagai berikut.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksa
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2))
>$float(%)
```

Hasil perhitungan analitik menunjukkan hasil yang sama dengan perhitungan sebelumnya.

Panjang Kurva

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

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

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \le t \le 2\pi.$$

```
>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):
```

Perintah diatas akan menghasilkan larik array t yang berjumlah 1000 nilai terdistribusi dari 0 sampai 2pi. Selajutnya dibuat plot dari korrdinat x dan y dengan skala yang diperbesar 1.5 kali.

```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t ...
                                                      ^
```

Sintaks diatas digunakan untuk menghitung panjang elemen diferensial pada kurva yaitu menghitung turunan pertama fungsi fx dan fy. Perintah "trigeduce" berfungsi untuk menyederhanakan ekspresi trigonometri dan perintah "radcan" untuk menyederhanakan ekspresi akar kuadrat.

```
>$integrate(ds(x),x,0,2*pi)
```

Pada pengintegralan fungsi diatas tidak bisa diselesaikan secara eksak oleh maxima sehingga kita harus menghitungnya secara numerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
```

```
  Function ds not found.
  Try list ... to find functions!
  Error in expression: ds(x)
   %mapexpression1:
      return expr(x,args());
  Error in map.
   %evalexpression:
      if maps then return %mapexpression1(x,f$;args());
  gauss:
      if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
  adaptivegauss:
      t1=gauss(f$,c,c+h;args(),=maps);
  Try "trace errors" to inspect local variables after errors.
  integrate:
      return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

Kita juga bisa menghitung panjang kurva yang berbentuk spiral logaritmik. Contoh fungsinya yaitu sebagai berikut.

$$x = e^{ax}\cos x,\ y = e^{ax}\sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
>&kill(a)
```

```
                                  done
```

Perintah diatas digunakan untuk menghapus variabel a.

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t
```

```
  Maxima said:
  diff: second argument must be a variable; found errexp1
   -- an error. To debug this try: debugmode(true);

  Error in:
  ... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t ...
                                                        ^
```

```
>S &=integrate(df(t),t,0,2*%pi); $S
```

```
Maxima said:
defint: variable of integration cannot be a constant; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
S &=integrate(df(t),t,0,2*%pi); $S ...
                               ^
```

Variabel S diatas menunjukan panjang kurva spiral.

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

```
Function S not found.
Try list ... to find functions!
Error in:
S(a=0.1) // Panjang kurva untuk a=0.1 ...
        ^
```

Bentuk kurva yang dapat kita hitung lagi dengan integral yaitu parabola.
Misalnya menunjukkan bahwa keliling lingkaran dengan jari-jari r adalah K=2.pi.r

```
>plot2d("x^2",xmin=-1,xmax=1):
```

FUNGSI MULTIVARIABEL

Sebuah fungsi multivariabel adalah pemetaan matematis yang menghubungkan beberapa variabel independen dengan satu variabel dependen. Fungsi ini umumnya dinotasikan sebagai

$$f(x,y) \text{ atau } f(x_1,x_2,...,x_n)$$

dimana $x,y,x_1,x_2,...,x_n$ adalah variabel independen

dan f adalah variabel dependen.

Fungsi multivariabel dapat diwakili dalam bentuk peta atau grafik tiga dimensi.
Contoh fungsi multivariabel :

$$z = x^2 + y^2$$

Dimana variabel bebasnya yaitu x dan y.

Grafik Fungsi Multivariabel

Pada fungsi multivariabel, grafik fungsinya merupakan grafik tiga dimensi. Ruang dimensi tiga dilambangkan dengan

$$R^3$$

Permukaan dalam R^3 terdapat dua macam yakni permukaan linear dan kuadratik. Setiap permukaan linear berupa bidang datar, sedangkan permukaan kuadratik berupa bidang lengkung yang kelengkungannya bergantung atas bentuk persamaannya.
Untuk membuat grafik fungsi tiga dimensi, maka kita dapat menggunakan perintah "plot3d()"

Grafik Fungsi Persamaan Linear Dalam Dimensi Tiga

Bentuk umum persamaan permukaan linear adalah

$$Ax + By + Cz + D = 0$$

Contoh grafik persamaan linear di ruang tiga dimensi

$$x - 1$$

```
>plot3d("x-1"):
>plot3d("3*x+4*y-12"):
>plot3d("7*x-1"):
```

Grafik Fungsi Kuadratik di Ruang Dimensi Tiga

Persamaan kuadratik mempunyai rumus umum :

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0$$

Permukaan-permukaan kuadratik dapat berupa permukaan bola, ellipsoida, paraboloida, tabung ellips, tabung lingkaran, atau tabung parabola.
Contoh grafik fungsi kuadratik di ruang dimensi tiga:

$$x^2 + y^2$$

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

Pada contoh diatas, digunakan ekspresi langsung dalam fungsi plot3d dan permukaan tersebut berupa paraboloida.

$$e^{-(x^2+y^2)}$$

```
>function f(x,y) := exp(-(x^2+y^2))
>plot3d("f", r=5):
```

Latihan:

$$y^2 - x^2$$

```
>plot3d("y^2-x^2",r=2,):
```

Gambar di atas merupakan paraboloida hiperbolik

Menggambar kurva perpotongan dari dua persamaan

Di EMT kita juga dapat menggabungkan dua kurva pada satu bidang untuk menggambarkan perpotongan. Untuk masalah ini kita gunakan fungsi >add dalam prosesnya.
Contoh 1:

$$x^2 + y^2 + z - 4 = 0$$

dengan

$$x^2 + y^2 = 1$$

```
>plot3d("x^2+y^2+z-4",r=5, implicit=3):
>plot3d("x^2+y^2-1",implicit=3, r=5, >add):
```

Dari persamaan diatas, kita dapatkan perpotongan antara paraboloida dengan tabung lingkaran.
contoh 2:

$$x^2 + y^2 + 2z = 9$$

dengan

$$y = 1$$

```
>plot3d("x^2+y^2+2z-9",r=5,implicit=3):
>plot3d("y-1",r=5, implicit=3, >add):
```

Dari persamaan diatas, didapatkan perpotongan antara bidang datar dan bidang lengkung.

Latihan:

$$x^2 + y^2 = 4$$

dengan

$$y^2 + z^2 = 4$$

```
>plot3d("x^2+y^2-4",r=5,implicit=3); plot3d("y^2+z^2-4",r=5,implicit=3,>add):
```

Didapatkan perpotongan antara dua tabung lingkaran.

Turunan Fungsi Multivariabel

Turunan Fungsi Dua Variabel

Turunan parsial, yaitu turunan fungsi terhadap satu variabel bebas sementara variabel bebas lainnya dianggap tetap atau konstan.
- Turunan Parsial terhadap f terhadap x di (x0,y0)

$$f_x(x_0, y_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x}$$

- Turunan Parsial f terhadap y di (x0,y0)

$$f_y(x_0, y_0) = \lim_{\Delta y \to 0} \frac{f(x_0, y_0 + \Delta y) - f(x_0, y_0)}{\Delta y}$$

Contoh soal untuk turunan parsial :

$$f(x, y) = 2xy - (1 - x^2)$$

```
>z &= 2*x*y-(1-x^2)
```

$$2 x y + x^2 - 1$$

```
>$showev('limit(((2*(x+h)*y)-(1-(x+h)^2))/h,h,0))
```

Perhitungan akan dilakukan menggunakan diff

```
>&diff(z,x) // z akan diturunkan terhadap x
```

$$2\ y\ +\ 2\ x$$

Karena pada fungsi z terdapat 2xy yang merupakan perkalian, untuk menghitung turunannya kita gunakan u'v+uv', sehingga turunan dari 2xy:

$$2.1.y + 2.x.0 = 2y$$

Kemudian turunan dari variabel berpangkat yaitu:

$$u^n = n.u^{n-1}.u'$$

Jadi turunan dari x^2:

$$x^2 = 2.x^{2-1}.1 = 2x$$

Turunan dari konstanta adalah 0
sehingga, turunan dari fungsi

$$f(x,y) = 2xy + x^2 - 1$$

terhadap x adalah

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

```
>&diff(z,y) // z akan diturunkan terhadap y
```

$$2\ x$$

Seperti pada turunan terhadap x, kita gunakan langkah yang sama namun kita anggap x konstan.

$$f(x,y) = 2xy + x^2 - 1$$

$$f_y(x,y) = 2.x.1 + 0 - 0$$

$$f_y(x,y) = 2x$$

Jadi, turunan terhadap y dari f(x,y) adalah 2x.
Sehingga, turunan parsial dari

$$f(x,y) = 2xy + x^2 - 1 \text{ adalah}$$

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

$$f_y(x,y) = 2x$$

Apabila kita gambarkan grafik untuk masing-masing turunan tersebut adalah sebagai berikut:

```
>plot3d("2*x+2*y"): // turunan terhadap x
>plot3d("2*x"): // turunan terhadap y
```

Integral

Integral Lipat Dua

Integral lipat dua f pada R, diberikan oleh

$$\int_a^b \int_c^d f(x,y)\,dx\,dy$$

atau bisa ditulis dengan

$$\int_R \int f(x,y)\,dA$$

Misalkan R adalah sebuah persegipanjang dengan sisi-sisi sejajar sumbu-sumbu koordinat; yaitu, misalkan

$$R = \{(x,y) : a \leq x \leq b, c \leq y \leq d\}$$

Urutan dx dan dy penting karena ia merinci integrasi mana yang akan dilakukan pertama.
SIFAT-SIFAT INTEGRAL LIPAT DUA:
1. Integral lipat dua bersifat linear, yaitu:

$$a. \iint_R kf(x,y)\,dA = k \iint_R f(x,y)\,dA$$

$$b. \iint_R [f(x,y) + g(x,y)]\,dA = \iint_R f(x,y)\,dA + \iint_R g(x,y)\,dA$$

2. Integral lipat dua bersifat aditif(dapat dijumlahkan) pada persegipanjang yang saling berhimpit pada hanya sebuah ruas garis

$$\iint_R f(x,y)\,dA = \iint_{R_1} f(x,y)\,dA + \iint_{R_2} f(x,y)\,dA$$

3. Sifat perbandingan berlaku jika

$$f(x,y) \leq g(x,y) \text{ untuk semua } (x,y) \text{ di R maka}$$

$$\iint_R f(x,y)\,dA \leq \iint_R g(x,y)\,dA$$

CONTOH :
1. Hitung

$$\iint (2xy)\,dx\,dy$$

Dengan memperhatikan urutan dx dan dy, pada soal kali ini kita hitung integralnya dari sebelah dalam yaitu kita integralkan terhadap x terlebih dahulu.

```
>qx &= integrate((2*x*y),x)
```

$$x^2 y$$

Setelah kita integralkan terhadap x, lalu hasilnya kita integralkan terhadap y.

```
>qy &= integrate(qx,y)
```

$$\frac{x^2 y^2}{2}$$

Jadi, hasil dari

$$\iint (2xy)\,dx\,dy$$

adalah

$$\frac{x^2y^2}{2} + c$$

2. Hitung

$$\int_0^3 \left[\int_1^2 (2x+3y)\,dx\right] dy$$

```
>f &= 2*x+3*y
```

```
3 y + 2 x
```

Dalam integral sebelah dalam, y berupa konstanta, sehingga

```
>fx &= integrate((2*x+3*y),x,1,2)
```

```
3 y + 3
```

Setelah mencari integral terhadap x, lalu kita integralkan hasil tersebut terhadap y, akibatnya

```
>fy &= integrate((3+3*y),y,0,3)
```

```
45
--
2
```

Aplikasi Fungsi Multivariabel

Penerapan Plot Kontur

Peta kontur seringkali digunakan untuk memperlihatkan kondisi cuaca atau lainnya dari berbagai titik dalam peta.

1.Buatlah plot kontur dari fungsi berikut

$f(x,y) = \frac{1}{2}(x^2+y^2)$ dengan level bilangan genap dari 0 sampai 20

```
>f &= 1/2*(x^2+y^2)
```

```
  2     2
 y   + x
 -------
    2
```

```
>aspect(1.5);
>plot2d(f,level=0:2:20,>hue,>spectral,n=200,grid=4,r=5):
```

2. Seorang ahli geologi sedang melakukan penelitian di daerah vulkanik. Dia tertarik untuk memahami bagaimana ketinggian permukaan tanah di sekitar gunung berapi berubah. Ketinggian di suatu titik (x, y) dalam kilometer di sekitar gunung berapi tersebut dapat dijelaskan oleh fungsi ketinggian H(x, y), di mana x dan y adalah koordinat dalam kilometer. Fungsi ketinggian H(x, y) diberikan oleh persamaan:

$$H(x, y) = 3x^2 - 2y^2$$

Gambarkan plot kontur dari fungsi ketinggian H(x, y) untuk memvisualisasikan perubahan ketinggian di sekitar gunung berapi dengan level ketinggian 1 sampai 7 km.

```
>h &= 3*x^2-2*y^2
```

$$3\,x^2 - 2\,y^2$$

```
>aspect(1.5);
>plot2d(h,level=1:1:7,>hue,>spectral,n=200,grid=4,r=2):
```

EMT Geometri

* Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan 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 tetapkan tiga titik dan plotlah.

```
>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 mencakup 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]
```

Hitunglah garis tegak lurus melalui A pada BC.

```
>h=perpendicular(A,lineThrough(B,C)): // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)): // D adalah titik potong h dan BC
```

Gambarkan itu.

```
>plotPoint(D,value=1): // koordinat D ditampilkan
>aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()
```

Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD.BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

```
  1.5
```

Compare with determinant formula.

```
>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 luar segitiga.

```
>c=circleThrough(A,B,C): // lingkaran luar segitiga ABC
>R=getCircleRadius(c): // jari2 lingkaran luar
>O=getCircleCenter(c): // titik pusat lingkaran c
>plotPoint(O,"O"): // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```

Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
  [1.16667,  1.16667]
  1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B): // garis bagi <ACB
>g=angleBisector(C,A,B): // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

```
  [0.86038,  0.86038]
```

Tambahkan semuanya ke dalam alur cerita.

```
>color(5); plotLine(l); plotLine(g); color(1): // gambar kedua garis bagi sudut
>plotPoint(P,"P"): // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
  0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gambar lingkaran dal
```

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(-2.5,4.5,-2.5,4.5); A=[-2,1]; plotPoint(A,"A");...
>B=[1,-2]; plotPoint(B,"B"); C=[4,4]; plotPoint(C,"C"):
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

```
>plotSegment(A,B,"c"); plotSegment(B,C,"a"); plotSegment(A,C,"b");...
>aspect(1):
```

3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

```
>l=angleBisector(A,C,B); g=angleBisector(C,A,B)
```

```
  [1.25658,  7.74342,  5.23025]
```

```
>P=lineIntersection(l,g)
```

```
  [0.581139,  0.581139]
```

```
>color(5); plotLine(l); plotLine(g); color(1); plotPoint(P,"P"):
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

```
  1.52896119631
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```

Jadi, terbukti bahwa garis bagi sudut yang ketiga juga melalui titik pusat lingkaran dalam.

4. Gambar jari-jari lingkaran dalam.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

```
  1.52896119631
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometry.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Akan tetapi, kita sekarang dapat menggunakan perhitungan simbolik.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan perhitungan simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

$$[-1, 2, 2]$$

Kita dapat memperoleh persamaan garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%,y) // persamaan garis melalu
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
>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
```

$$\left[\frac{2}{5}, \frac{6}{5}\right]$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
>$distance(A,Q) // jarak AQ
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
>$computeAngle(A,C,B) // nilai <ACB
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 ga
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
  [0.86038,  0.86038]
```

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat membuat garis berpotongan dengan lingkaran, dan lingkaran dengan lingkaran.

```
>A &:= [1,0]; c=circleWithCenter(A,4)
```

```
  [1,  0,  4]
```

```
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C)
```

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

```
>setPlotRange(5); plotCircle(c); plotLine(l):
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>a{P1,P2,f}&=lineCircleIntersections(l,c):
```

```
  } missing!
  Error in:
  a{P1,P2,f}&=lineCircleIntersections(l,c): ...
         ^
```

```
>P1, P2, f
```

```
  Variable P1 not found!
  Error in:
  P1, P2, f ...
    ^
```

```
>plotPoint(P1); plotPoint(P2):
```

```
  Variable or function P1 not found.
  Error in:
  plotPoint(P1); plotPoint(P2): ...
              ^
```

Sama halnya 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
```

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):
```

```
  Variable or function P1 not found.
  Error in:
  ... ormalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegm ...
                                                              ^
```

```
>degprint(computeAngle(P1,C,P2))
```

```
  Variable or function P1 not found.
  Error in:
  degprint(computeAngle(P1,C,P2)) ...
                           ^
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C):
```

```
  Variable or function P1 not found.
  Error in:
  ... ormalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegm ...
                                                        ^
```

```
>degprint(computeAngle(P1,C,P2))
```

```
  Variable or function P1 not found.
  Error in:
  degprint(computeAngle(P1,C,P2)) ...
                           ^
```

```
>insimg;
```

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):
```

Selanjutnya, kita 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 perpotongan cukup rumit. Namun, kita dapat menyederhanakannya, jika kita mencari nilai y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sepenuhnya berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

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:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

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:

$$x^2+y^2=b^2, \quad (x-a)^2+y^2=c^2.$$

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)");  ...
>plotPoint([7.5,6], "A(x,y)"):
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15);  ...
>plotSegment([0,0],[7.5,6],"b",25):
>plotSegment([7.5,6],[7.5,0],"t=y",25):
>&assume(a>0); sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

[]

Ekstrak solusi y.

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

```
  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]; $'y=sqrt(factor(ysol^2)) ...
                            ^
```

Kita mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

Tentu saja, setiap segitiga siku-siku 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 a*abs(ysol)/2
  Error in:
  H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
          ^
```

Dan jelas pula, bahwa ini adalah segitiga dengan luas maksimal dan dua 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)mxb
```

```
Maxima said:
incorrect syntax: mxb is not an infix operator
a,b,c)^2,c)=0,c)mxb;
                ^

Error in:
 $solve(diff(H(a,b,c)^2,c)=0,c)mxb ...
                              ^
```

Sekarang mari kita cari himpunan semua titik di mana b+c=d untuk suatu 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 buat fungsi ini.

```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d
```

Sekarang kita dapat menggambar himpunannya. 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):
```

Kita dapat memeriksa persamaan umum untuk 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 sumbu setengah.

```
>$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])
```

Kita melihat bahwa tinggi dan luas segitiga tersebut adalah maksimum untuk x=0. Jadi luas segitiga dengan a+b+c=d adalah maksimum, jika segitiga tersebut sama sisi. Kita ingin memperolehnya secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

Kita memperoleh beberapa nilai minimum, yang dimiliki oleh segitiga dengan satu sisi 0, dan solusinya a=b=c=d/3.

```
>$solve(eqns,[a,b])
>&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 dari situasinya
Pertama-tama atur titik 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
```

Kemudian atur rentang plot dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
>plotPoint(mxmeval("A"),"A"):
```

```
  Variable a1 not found!
  Use global variables or parameters for string evaluation.
  Error in Evaluate, superfluous characters found.
  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 segmen.

```
>splotSegment(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:
  splotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval(" ...
                              ^
```

Hitunglah garis tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat kelilingnya.

```
>U &= lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran luar.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

Mari kita tambahkan ini ke dalam alur cerita.

```
>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 ...
                ^
```

Dengan menggunakan geometri, kita peroleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk jari-jari. Kita dapat memeriksa apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))^2  | factor
```

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Garis ini merupakan garis pusat segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga tersebut, termasuk orthocenter, circumcenter, centroid, titik Exeter, dan titik pusat lingkaran sembilan titik pada segitiga tersebut.

Sebagai contoh, kita hitung dan plot garis Euler dalam sebuah segitiga.

Pertama, kita definisikan sudut-sudut segitiga dalam Euler. Kita gunakan definisi, yang terlihat dalam ekspresi simbolik.

```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titik ke dalamnya. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kita juga dapat menambahkan sisi-sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
```

Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

$$[-1, 3, -2]$$

Dan dapatkan juga rumus untuk garis ini.

```
>$getLineEquation(c,x,y)
```

Untuk bentuk Hesse, kita perlu menentukan suatu titik, sehingga titik tersebut berada di sisi positif bentuk Hesse. Memasukkan titik akan menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
>O &= getCircleCenter(LL); $O
```

Gambarkan lingkaran dan titik pusatnya. Cu dan U adalah simbol. Kita evaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```

Kita dapat menghitung perpotongan tinggi di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
> perpendicular(B,lineThrough(A,C))); $H
```

Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

Tambahkan ke plot kita.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

Pusat gravitasi seharusnya berada pada garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
>plotPoint(M(),"M"): // titik berat
```

Teori ini memberi tahu kita MH=2*MO. Kita perlu menyederhanakannya dengan radcan untuk mencapainya.

```
>$distance(M,H)/distance(M,O)|radcan
```

Fungsinya termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

```
  60°15’18.43’’
```

Persamaan untuk pusat lingkaran dalam tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

Mari kita hitung juga ekspresi untuk jari-jari lingkaran dalam.

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
>LD &=  circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke dalam alur cerita.

```
>color(5); plotCircle(LD()):
```

Parabola

Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```

Ini seharusnya merupakan suatu fungsi, tetapi penyelesai default Maxima hanya dapat menemukan solusinya, jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

```
>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]
```

Solusi pertama adalah

maxima: akar[1]

Dengan menambahkan solusi pertama ke dalam plot, terlihat bahwa itu memang jalur yang kita cari. Teori tersebut memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jatak T ke AB
```

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

Hal ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti konsep klasik jarak dan sudut dengan kuadran dan sebaran. Dengan menggunakan konsep ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan simbolik rasional sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga siku-siku dengan proporsi Mesir yang terkenal yaitu 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);
```

Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana wa adalah sudut di A. Cara yang biasa untuk menghitung sudut ini adalah dengan mengambil kebalikan 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.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Faktanya, itu hanyalah kuadrat jarak. Dalam persamaan berikut, a, b, dan c menunjukkan kuadran sisi-sisi.

Teorema Pythogoras menjadi a+b=c.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Gagasan kedua trigonometri rasional adalah sebaran. Sebaran mengukur bukaan antara garis. Nilainya 0, jika garis sejajar, dan 1, jika garis persegi panjang. Nilainya adalah kuadrat sinus sudut antara dua garis.

Sebaran garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadran dari setiap segitiga persegi panjang dengan satu sudut di A.

```
>sa &= a/c; $sa
```

Tentu saja, ini lebih mudah dihitung daripada sudut. Namun, Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengubah nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

```
  9/25
```

Hukum kosinus dari trgonometri klasik diterjemahkan ke dalam "hukum silang" berikut.

$$(c+b-a)^2 = 4bc\,(1-s_a)$$

Di sini a, b, dan c adalah kuadran sisi-sisi segitiga, dan sa adalah sebaran di sudut A. Sisi a, seperti biasa, berseberangan dengan sudut A.

Hukum-hukum ini diimplementasikan dalam berkas geometry.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

Mari kita gunakan hukum silang ini untuk menemukan sebaran di A. Untuk melakukannya, kita buat hukum silang untuk kuadran a, b, dan c, dan selesaikan untuk sebaran yang tidak diketahui sa.

Anda dapat melakukannya dengan mudah secara manual, tetapi saya menggunakan Maxima. Tentu saja, kita mendapatkan hasil yang sudah kita miliki.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

Kita sudah tahu ini. Definisi sebaran adalah kasus khusus dari hukum silang.

Kita juga dapat memecahkan ini untuk a,b,c umum. Hasilnya adalah rumus yang menghitung sebaran sudut segitiga yang diberikan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

Kita dapat membuat fungsi dari hasil tersebut. Fungsi tersebut telah didefinisikan dalam berkas geometry.e milik Euler.

```
>$spread(a,b,c)
```

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi-sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

```
  67°47'32.44''
```

Contoh Lain

Sekarang, mari kita coba contoh yang lebih maju.

Kita tentukan 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;
```

Dengan menggunakan Pythogoras, mudah untuk menghitung jarak antara dua titik. Pertama-tama saya menggunakan fungsi distance dari file Euler untuk geometri. Fungsi distance menggunakan geometri klasik.

```
>$distance(A,B)
```

Euler juga memuat fungsi untuk kuadran antara dua titik.
Dalam contoh berikut, karena c+b bukan a, segitiga tersebut bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan perkalian titik dua vektor. Hasilnya adalah beberapa perkiraan floating point.

$$A =< 1,2 > \quad B =< 4,3 >, \quad C =< 0,4 >$$

$$\mathbf{a} = C - B =< -4,1 >, \quad \mathbf{c} = A - B =< -3,-1 >, \quad \beta = \angle ABC$$

$$\mathbf{a}.\mathbf{c} = |\mathbf{a}|.|\mathbf{c}| \cos\beta$$

$$\cos\angle ABC = \cos\beta = \frac{\mathbf{a}.\mathbf{c}}{|\mathbf{a}|.|\mathbf{c}|} = \frac{12-1}{\sqrt{17}\sqrt{10}} = \frac{11}{\sqrt{17}\sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

```
  32.4711922908
```

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kita masukkan kuadran a, b, dan c ke dalam hukum silang dan selesaikan untuk x.

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

Yaitu, apa yang dilakukan fungsi sebaran yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

Maxima memperoleh hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Maxima memang menyelesaikan suku sin(arccos(...)) menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

Setelah kita memiliki sebaran di B, kita dapat menghitung tinggi ha pada sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

menurut definisi.

```
>ha &= c*sb; $ha
```

Gambar berikut ini dibuat dengan program geometri C.a.R., yang dapat menggambar kuadran dan sebaran.
image: (20) Rational_Geometry_CaR.png
Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

```
>$sqrt(ha)
```

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, bahwa kita sedang membahas tentang kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

Rumus determinan yang biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama-tama kita hitung sebaran di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita hitung luas kuadrat (yang disebut "quadrea"?), faktorkan dengan Maxima, dan kita dapatkan 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)
```

Aturan Triple Spread

Kerugian spread adalah tidak lagi sekadar menambahkan sudut yang sama.
Namun, tiga spread segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

Aturan ini berlaku untuk tiga sudut yang jumlahnya mencapai 180°.

$$\alpha+\beta+\gamma=\pi$$

Karena sebaran

$$\alpha,\pi-\alpha$$

sama, aturan sebaran rangkap tiga juga berlaku, jika

$$\alpha+\beta=\gamma$$

Karena sebaran sudut negatif sama, aturan sebaran rangkap tiga juga berlaku, jika

$$\alpha+\beta+\gamma=0$$

Misalnya, kita dapat menghitung sebaran sudut 60°. Yaitu 3/4. Persamaan tersebut memiliki solusi kedua, di mana semua sebarannya adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

Sebaran 90° jelas adalah 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya memecahkan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita memperoleh a+b=1.

```
>$triplespread(x,y,1), $solve(%,x)
```

Karena sebaran 180°-t sama dengan sebaran t, rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dari dua sudut lainnya.
Jadi kita dapat menemukan sebaran sudut yang digandakan. Perhatikan bahwa ada dua solusi lagi. Kita buat ini menjadi fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

```
                                - 4 (a - 1) a
```

Garis Bagi Sudut

Kita sudah tahu situasinya seperti ini.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```

Mari kita hitung panjang garis bagi sudut di A. Namun, kita ingin menyelesaikannya untuk a,b,c umum.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung sebaran sudut yang dibagi dua di A, menggunakan rumus sebaran rangkap tiga.
Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut yang dibagi dua 180°-wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

Kita dapat mencetak sudut dalam Euler, setelah mentransfer sebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

```
  18°26’5.82’’
```

Titik P merupakan perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
  [0,  1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```

Mari kita periksa sudut-sudut pada contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
  0.321750554397
  0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.
Kita gunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku di sembarang segitiga. Kuadratkan, maka akan menghasilkan apa yang disebut "hukum sebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a,b,c menunjukkan kuadran.
Karena CPA sebaran adalah 1-sa2, kita peroleh darinya bisa/1=b/(1-sa2) dan dapat menghitung bisa (kuadran garis bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
  4.21637021356
  4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
  1.33333333333
```

Sudut Tali

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;
```

Kita dapat menggunakan Maxima untuk memecahkan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Dengan demikian, kita memperoleh rumus untuk jari-jari kuadrat pericircle dalam bentuk kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

Kita dapat menjadikannya fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk titik A, B, C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

```
>periradius(a,b,c)
```

```
  1
```

Faktanya, sebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut tali busur.

```
>$spread(b,a,c)*rabc | ratsimp
```

Faktanya, sebarannya adalah b/(4r), dan kita melihat bahwa sudut tali busur b adalah setengah sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

Fungsi yang, pada titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis datar yang agak sederhana: lingkaran yang berpusat di A.

```
>&remvalue();
>A=[-1,-1];
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
>title="If you see ellipses, please set your window square"):
```

dan grafiknya cukup sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

Tentu saja minimum 0 dicapai di A.

Dua titik

Sekarang kita lihat fungsi MA+MB di mana A dan B adalah dua titik (tetap). Merupakan "fakta yang diketahui" 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):
```

Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

Tiga poin

Sekarang semuanya menjadi kurang sederhana: Tidak banyak yang tahu bahwa MA+MB+MC mencapai nilai minimumnya di satu titik bidang, tetapi menentukannya tidaklah sesederhana itu:

1) Jika salah satu sudut segitiga ABC lebih dari 120° (misalkan di A), maka nilai minimumnya tercapai di titik ini (misalkan 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;
>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;
```

2) Namun jika semua sudut segitiga ABC kurang dari 120°, maka nilai minimumnya berada di titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang sudut-sudut sisi ABC-nya sama (masing-masing sudutnya 120°):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
>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;
```

Merupakan aktivitas yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya mengetahui perangkat lunak yang ditulis dalam Java yang memiliki instruksi "garis kontur"...

Semua ini ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; ia menulis surat kepada para dilettan lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sehingga FA+FB+FC minimal, disebut titik Fermat dari segitiga tersebut. Namun tampaknya beberapa tahun sebelumnya, Torriccelli dari Italia telah menemukan titik ini sebelum Fermat menemukannya! Bagaimanapun tradisinya adalah mencatat titik F ini...

Empat titik

Langkah berikutnya adalah menambahkan titik ke-4 D dan mencoba meminimalkan MA+MB+MC+MD; katakanlah Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena Anda sehingga Anda dapat menyalurkan sinyal ke empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
>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;
```

Masih terdapat nilai minimum dan tidak tercapai di titik A, B, C, maupun 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 bersifat rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kita mengharapkan bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
>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;
```

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe di jalur program.

Pertama, kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita memerlukan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometry.e milik Euler untuk ini.

```
>load geometry;
```

Pertama dua garis 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])
```

$$[-1, 2, 1]$$

Kita merencanakan segalanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
>color(black); plotLine(g(),"")
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```

Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

$$[0, u]$$

Hitunglah jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

Hitunglah jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

Dan temukan pusat kedua lingkaran, yang jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol
```

Ada dua solusi.
Kita mengevaluasi solusi simbolik, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()
```

```
  [0.333333,  1]
```

```
>dd := d()
```

```
  [0.149071,  0.447214]
```

Gambarkan lingkaran-lingkaran tersebut ke dalam gambar.

```
>color(red); plotCircle(circleWithCenter([0,u[1]],dd[1]),""); ...
>plotCircle(circleWithCenter([0,u[2]],dd[2]),""); insimg;
```

Plot dengan Povray

Selanjutnya kita plot semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.
Pertama-tama 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 menyiapkan suasananya dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Berikutnya kita menulis kedua bola itu ke dalam 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)));
```

Kita buat bidang yang dibatasi 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 kita buat 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]]);
```

Kemudian kita buat dua titik tempat bola-bola tersebut menyentuh bidang. Titik-titik 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)));
```

Berikutnya kita hitung 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)));
```

Kita menghubungkan titik-titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang kita buat pita abu-abu, di mana bola-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 () ...
```

```
  endfunction
```

Anda memerlukan kacamata merah/cyan untuk menghargai efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°):
```

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam berkas "spherical.e" di folder contoh. Kami perlu memuat berkas tersebut terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut ini adalah koordinat untuk 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 bulat).

```
>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 hitung vektor dari satu ke yang lain pada bola ideal. Vektor ini adalah [arah, jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan 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. Rutin berikut menggunakan perkiraan yang lebih baik lagi. Pada jarak yang pendek, hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km"; // perkiraan jarak FMIPA-Semarang
```

Ada fungsi untuk judul, yang memperhitungkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
       65.34°
```

Sudut suatu segitiga melebihi 180° 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 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
```

Kita juga dapat menambahkan vektor ke posisi. Vektor berisi arah dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kita menggunakan svector. Untuk menambahkan vektor ke posisi, kita 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 bentuk bola yang ideal. Sama halnya 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 lihat contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
>sposprint(Tugu), sposprint(Monas)
```

```
  S 7°46.998’ E 110°21.966’
  S 6°10.500’ E 106°48.717’
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami memperoleh perkiraan yang baik.

```
>esdist(Tugu,Monas)->" km"; // perkiraan jarak Tugu Jogja - Monas Jakarta
```

Judulnya sama dengan yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
  294°17’2.85’’
```

Akan tetapi, kita tidak lagi memperoleh posisi target yang tepat, jika kita menambahkan arah dan jarak ke posisi awal. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan radius bumi di sepanjang lintasan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
  S 6°10.500’ E 106°48.717’
```

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

```
  S 6°10.500’ E 106°48.717’
```

Tentu saja, kita tidak dapat berlayar dengan arah yang sama dari satu tujuan ke tujuan lain, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke arah timur laut 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 kita jauh dari tujuan yang benar, jika kita menggunakan arah yang sama selama perjalanan kita.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 dikalikan sepersepuluh jaraknya, dengan memakai arah ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya sangat 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 garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Lintasan terpendek dari P1 ke P2 bukanlah lingkaran lintang 30°, tetapi lintasan yang lebih pendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
       79.69°
```

Namun, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kita menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1;  dist=esdist(P1,P2); ...
>loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
       79.69°
       81.67°
       83.71°
       85.78°
       87.89°
       90.00°
       92.12°
       94.22°
       96.29°
       98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti arah yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
       0.203km
```

Kita memperoleh perkiraan yang baik, jika kita menyesuaikan arah setelah setiap 1/100 jarak total 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 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'
```

Kita menulis suatu fungsi yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
  $useglobal;
  endfunction
```

Sekarang rencanakan semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```

Atau gunakan plot3d untuk mendapatkan tampilan anaglifnya. Ini tampak sangat bagus dengan kaca mata merah/biru kehijauan.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```

Latihan

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah (360/n).
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan (360/n).
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

```
>load geometry
```

```
  Numerical and symbolic geometry.
```

```
>setPlotRange(-3.5,3.5,-3.5,3.5);
>A=[-2,-2]; plotPoint(A,"A");
>B=[2,-2]; plotPoint(B,"B");
>C=[0,3]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
>aspect(1):
>c=circleThrough(A,B,C); R=getCircleRadius(c); O=getCircleCenter(c);...
>plotPoint(O,"O"); l=angleBisector(A,C,B); color(2); plotLine(l); color(1);...
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya y= ax^2+bx+c.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

```
>load geometry;
>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R"):
>sol &= solve([a+b=-c,16*a+4*b=-c,c=-4],[a,b,c])
```

```
                    [[a = - 1, b = 5, c = - 4]]
```

```
>function y&=-x^2+5*x-4
```

```
                    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.

```
>load geometry
```

```
  Numerical and symbolic geometry.
```

```
>setPlotRange(-4.5,4.5,-4.5,4.5);
>A=[-3,-3]; plotPoint(A,"A");
>B=[3,-3]; plotPoint(B,"B");
>C=[3,3]; plotPoint(C,"C");
>D=[-3,3]; plotPoint(D,"D");
>plotSegment(A,B,"");
>plotSegment(B,C,"");
>plotSegment(C,D,"");
>plotSegment(A,D,"");
>aspect(1):
>l=angleBisector(A,B,C); m=angleBisector(B,C,D); P=lineIntersection(l,m);...
>color(5); plotLine(l); plotLine(m); color(1); plotPoint(P,"P"):
```

Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)));
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD"):
```

Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat.
Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

```
>AB=norm(A-B) //panjang sisi AB
```

```
  6
```

```
>CD=norm(C-D) //panjang sisi CD
```

```
  6
```

```
>AD=norm(A-D) //panjang sisi AD
```

```
  6
```

```
>BC=norm(B-C) //panjang sisi BC
```

```
  6
```

```
>AB.CD, AD.BC
```

```
  36
  36
```

Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

Penyelesaian :
Diketahui kedua titik fokus P = [-1,-1] dan Q = [1,-1]

```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x-Q[1])^2+(y-Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
>reset()
```

```
  0
```

Grafik yang lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

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).

```
>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):
>reset()
```

```
  0
```

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
>reset()
```

```
  0
```

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

EMT untuk Statistika

Di 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 latar belakang untuk memahami detailnya.

Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
>median(M), mean(M), dev(M),
```

```
  999
  999.9
  2.72641400622
```

Kita dapat memplot plot kotak-dan-kumis untuk datanya. Dalam kasus kami, tidak ada outlier.

```
>aspect(1.75); boxplot(M):
```

Kami menghitung probabilitas suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur berdistribusi 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 print.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

```
        3.07 %
```

Untuk contoh berikutnya, kita asumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah alur pendistribusiannya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\/"):
```

Kita bisa memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah pria 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=["BB","BA","Frek"])
```

```
          BB        BA      Frek
       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 memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang tersebut. Kita bisa menggunakan dua kolom pertama tabel kita untuk ini.

Sumbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" 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
```

Namun akan lebih mudah jika menjumlahkan 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 mean dan deviasi sampel dengan frekuensi tertentu.

```
>{m,d}=meandev(M,v); m, d,
```

```
  169.901234568
  5.98912964449
```

Mari kita tambahkan distribusi nilai normal ke diagram batang di atas. Rumus distribusi normal dengan mean m dan simpangan baku d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{\frac{-(x-m)^2}{2d^2}}.$$

Karena nilainya antara 0 dan 1, maka untuk memplotnya pada bar plot harus dikalikan dengan 4 kali jumlah data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
>  xmin=min(r),xmax=max(r),thickness=3,add=1):
```

Tabel

Di direktori buku catatan ini Anda menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file tersebut. Datanya berasal dari buku online Jerman "Einführung in die Statistik mit R" 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 tokennya.

Untuk ini, kami mendefinisikan kumpulan token. Fungsi strtokens() mendapatkan vektor string token dari string tertentu.

```
>mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
```

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen tok2, tok4 dll. adalah terjemahan dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda perlu 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 kumpulan token yang sama. Kami mencetak empat baris pertama saja.

```
>writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
  MT is not a variable!
  Error in:
  writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,to ...
                     ^
```

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token yang akan diterjemahkan 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() kini mengembalikan sekumpulan token.

```
>tok
```

```
  Variable tok not found!
  Error in:
  tok ...
     ^
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke dalam angka.
String khusus NA = "." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
>MT[1]
```

```
  MT is not a variable!
  Error in:
  MT[1] ...
       ^
```

Berikut isi tabel dengan nomor yang belum diterjemahkan.

```
>writetable(MT,wc=5)
```

```
  Variable or function MT not found.
  Error in:
  writetable(MT,wc=5) ...
              ^
```

Untuk kenyamanan, Anda dapat memasukkan keluaran 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;
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. atau menggunakan tabel daftar.

```
>writetable(Table,ctok=ctok,wc=5);
```

```
  Variable or function Table not found.
  Error in:
  writetable(Table,ctok=ctok,wc=5); ...
                 ^
```

Fungsi tablecol() mengembalikan nilai kolom tabel, melewatkan baris apa pun 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 bisa menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

```
  MT is not a variable!
  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 suatu 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, ...
                                          ^
```

Kita bisa mencetak hasilnya di 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 v pada 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 diekstraksi dan diurutkan di kolom ke-5.

```
>writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);
```

```
  MT1 is not a variable!
  Error in:
  writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7); ...
                   ^
```

Untuk statistik selanjutnya, kami ingin menghubungkan dua kolom tabel. Jadi kita ekstrak kolom 2 dan 4 dan urutkan tabelnya.

```
>i=sortedrows(MT,[2,4]);  ...
>  writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

```
  Variable or function MT not found.
  Error in:
  i=sortedrows(MT,[2,4]);    writetable(tablecol(MT[i],[2,4])',c ...
                  ^
```

Dengan getstatistics(), kita juga bisa menghubungkan jumlah dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...
>{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
>writetable(count,labr=tok[xu1],labc=tok[xu2])
```

```
  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 tersebut.

```
>{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 file tersebut.

```
>fileremove(filename);
```

Distribusi

Dengan plot2d, ada metode yang sangat mudah untuk memplot sebaran 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
```

Perlu diperhatikan perbedaan antara bar plot (sampel) dan kurva normal (distribusi sebenarnya). Masukkan kembali ketiga perintah untuk melihat hasil pengambilan sampel lainnya.

Berikut adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta outlier.

```
>p=normal(10,1000); boxplot(p):
```

Untuk menghasilkan bilangan bulat acak, Euler memiliki intrandom. Mari kita simulasikan lemparan dadu dan plot distribusinya.

Kita menggunakan fungsi getmultiplicities(v,x), yang menghitung seberapa sering elemen v muncul di x. Kemudian kita plot hasilnya menggunakan kolomplot().

```
>k=intrandom(1,6000,6);  ...
>columnsplot(getmultiplicities(1:6,k));  ...
>ygrid(1000,color=red):
```

Meskipun inrandom(n,m,k) mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 hingga k, distribusi bilangan bulat lainnya dapat digunakan dengan randpint().

Dalam contoh berikut, probabilitas untuk 1,2,3 masing-masing adalah 0,4,0.1,0.5.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
  [395,  106,  499]
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihat referensinya.

Misalnya, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan berdistribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

$$\lambda = \frac{1}{\mu}, \quad \mu \text{ adalah mean, dan dilambangkan dengan } X \sim \text{Exponential}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```

Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

```
>plot2d("normaldis",-4,4):
```

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):
```

$$\text{normaldis(x,m,d)} = \int_{-\infty}^{x} \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2}\, dt.$$

Peluang berada di kawasan 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 yang mean dan deviasinya sama. Fungsi invbindis() menyelesaikan 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 kepadatan distribusi chi-kuadrat. Seperti biasa, Euler memetakan 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):
```

Euler memiliki fungsi akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.

Penamaannya mencoba untuk konsisten. Misalnya.,

- distribusi chi-kuadratnya adalah chidis(),
- fungsi kebalikannya adalah invchidis(),
- kepadatannya adalah qchidis().

Pelengkap 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 hampir merupakan distribusi yang seragam. Mari kita tentukan generator nomor acak untuk ini. Fungsi find(v,x) mencari nilai x pada 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 banyak iterasi.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```

Berikut adalah fungsi sederhana untuk memeriksa keseragaman distribusi nilai 1...K dalam v. Kita 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 fungsinya menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

```
  0
```

Dan ia 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 defaultnya adalah alfa.

Arti dari 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. Namun untuk n yang besar, penjumlahan langsungnya tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
  0.751401349655
```

invbinsum() menghitung kebalikan dari binomialsum().

```
>invbindis(0.75,1000,0.4)
```

```
  409.932733047
```

Di Bridge, kami mengasumsikan 5 kartu 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])
```

```
            406             80            514
            419             92            489
            413            110            477
            396             88            516
            385            107            508
            366             93            541
            399             98            503
            431             74            495
            398            111            491
            395            116            489
```

Merencanakan Data

Untuk memetakan data, kami mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam jumlah kursi.

```
>BW := [ ...
>1990,662,319,239,79,8,17; ...
>1994,672,294,252,47,49,30; ...
>1998,669,245,298,43,47,36; ...
>2002,603,248,251,47,55,2; ...
>2005,614,226,222,61,51,54; ...
>2009,622,239,146,93,68,76; ...
>2013,631,311,193,0,63,64];
```

Untuk pesta, kami menggunakan rangkaian nama.

```
>P:=["CDU/CSU","SPD","FDP","Gr","Li"];
```

Mari kita cetak persentasenya dengan baik.

Pertama kita mengekstrak kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi seluruhnya. kolom adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian statistiknya kita cetak dalam bentuk tabel. Kami menggunakan nama sebagai header kolom, dan tahun sebagai header untuk baris. Lebar default untuk kolom adalah wc=10, tetapi kami lebih memilih keluaran 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 ini menjumlahkan persentase dua partai besar yang menunjukkan bahwa partai-partai kecil berhasil memperoleh suara di parlemen hingga tahun 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"):
```

Tentukan beberapa warna untuk setiap pesta.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

Sekarang kita bisa memplot hasil pemilu 2009 dan perubahannya menjadi satu plot dengan 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):
```

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):
```

Plot kolom 3D memperlihatkan baris data statistik dalam bentuk kolom. Kami memberikan label untuk baris dan kolom. sudut adalah sudut pandang.

```
>columnsplot3d(BT,scols=P,srows=YT, ...
>  angle=30°,ccols=CP):
```

Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kami mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller);  ...
>mosaicplot(BT',srows=YT,scols=P,color=CP,style="#"); ...
>shrinkwindow():
```

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]):
```

Ini adalah jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```

Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls dari data acak, terdistribusi secara seragam di [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```

Namun untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```

Fungsi Columnplot() lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kami telah mendemonstrasikannya di tutorial ini.

Ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat 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):
```

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"]):
```

Berikut ini cara memplot frekuensi bilangan dalam suatu vektor.
Kami membuat vektor bilangan acak bilangan bulat 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

```
  [2,  3,  7,  10,  3,  9,  10,  5,  1,  5]
```

Kemudian ekstrak nomor unik di v.

```
>vu:=unique(v)
```

```
  [1,  2,  3,  5,  7,  9,  10]
```

Dan plot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```

Kami ingin mendemonstrasikan fungsi distribusi nilai empiris.

```
>x=normal(1,20);
```

Fungsi empdist(x,vs) memerlukan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

```
>xs=sort(x);
```

Kemudian kita plot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi kali ini kami menggunakan plot gigi gergaji.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar));  ...
>figure(0):
```

Plot sebar 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=".."):
```

Seringkali kita ingin membandingkan dua sampel dengan distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujiannya, 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):
```

Plot tersebut dengan jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.

Jika kita mempunyai dua distribusi yang ukurannya berbeda, kita dapat memperluas distribusi yang lebih kecil atau mengecilkan distribusi yang lebih besar. Fungsi berikut ini 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):
```

Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi fit.

Sebagai permulaan kita menemukan garis regresi untuk data univariat dengan polyfit(x,y,1).

```
>x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x'|y',labc=["x","y"])
```

x	y
1	2
2	3
3	1
4	5

```
  5         6
  6         3
  7         7
  8         8
  9         9
 10         8
```

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. 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):
```

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 alih-alih membiarkan readtable() mengumpulkan terjemahannya.

```
>{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 muncul dari plot sebar 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 terlihat bahwa usia ayah dan ibu saling 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 merupakan model yang salah. Garis regresinya adalah s=17+0,74t, dengan t adalah umur ibu dan s adalah umur ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tapi tidak terlalu banyak.

Sebaliknya, kami mencurigai 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 sebar.

```
>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 tersebut. Ini hanya menunjukkan, bahwa usianya berbeda-beda.

```
>boxplot(cs,["mothers","fathers"]):
```

```
  Variable or function cs not found.
  Error in:
  boxplot(cs,["mothers","fathers"]): ...
            ^
```

Menariknya, perbedaan median tidak sebesar perbedaan mean.

```
>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 pangkat merupakan ukuran keteraturan yang sama pada kedua vektor. Hal 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))
```

```
  -0.500500593945
```

Berikut adalah fungsi lainnya, yang disebut koefisien skewness Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

```
  -0.00473573537442
```

Simulasi Monte Carlo

Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Ini satu lagi, yang mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan pembagian jumlahnya.

```
>ds:=sum(intrandom(1000,3,6))';  fs=getmultiplicities(3:18,ds)
```

```
  [4,  17,  25,  40,  52,  85,  118,  145,  130,  124,  110,  70,  39,
  24,  14,  3]
```

Kita bisa merencanakannya sekarang.

```
>columnsplot(fs,lab=3:18):
```

Untuk menentukan distribusi yang diharapkan tidaklah mudah. Kami menggunakan rekursi tingkat lanjut untuk ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat direpresentasikan sebagai jumlah dari 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 ini hasil dari tiga kali lemparan dadu.

```
>countways(5:25,5,5)
```

```
  [1,  5,  15,  35,  70,  121,  185,  255,  320,  365,  381,  365,  320,
  255,  185,  121,  70,  35,  15,  5,  1]
```

```
>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):
```

Untuk simulasi lain, deviasi nilai rata-rata dari n variabel acak berdistribusi normal 0-1 adalah 1/akar(n).

```
>longformat; 1/sqrt(10)
```

```
  0.316227766017
```

Mari kita periksa ini dengan simulasi. Kita hasilkan 10000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M)')
```

```
  0.313795259535
```

```
>plot2d(mean(M)',>distribution):
```

Median dari 10 bilangan acak berdistribusi normal 0-1 memiliki deviasi yang lebih besar.

```
>dev(median(M)')
```

```
  0.368587928611
```

Karena kita dapat dengan mudah menghasilkan lintasan acak, kita dapat mensimulasikan proses Wiener. Kita mengambil 1000 langkah dari 1000 proses. Kemudian kita memetakan deviasi standar dan rata-rata 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):
```

Pengujian

Pengujian merupakan alat penting dalam statistik. Dalam Euler, banyak pengujian yang diterapkan. Semua pengujian ini menghasilkan galat yang kita terima jika kita menolak hipotesis nol.

Sebagai contoh, kita menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kita memperoleh nilai berikut, yang kita masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

```
  0.498830517952
```

Uji chi-square juga memiliki modus, 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)
```

```
  0.509
```

Kesalahan ini terlalu besar. Jadi kita tidak dapat menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Namun, kita tidak dapat menolak hipotesis kita.

Selanjutnya, kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan pengujian yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

```
  0.753754444313
```

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

```
  0.275383759442
```

Fungsi ttest() memerlukan nilai rata-rata, deviasi, jumlah data, dan nilai rata-rata yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk nilai rata-rata yang sama. Kita tolak hipotesis bahwa keduanya memiliki nilai rata-rata yang sama, jika hasilnya <0,05.

```
>tcomparedata(normal(1,10),normal(1,10))
```

```
  0.330456077233
```

Jika kita menambahkan bias pada satu distribusi, kita akan mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

```
  5.97533459989e-06
```

Pada contoh berikutnya, kita buat 20 lemparan dadu acak sebanyak 100 kali dan hitung angka-angka yang ada di dalamnya. Rata-rata harus ada 20/6=3,3 angka.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

```
  3.2
```

Sekarang kita bandingkan jumlah angka satu dengan distribusi binomial. Pertama kita gambarkan distribusi angka satu.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\/"):
>t=count(R,21);
```

Lalu kami hitung 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)
```

```
  0.697496276422
```

Contoh berikut berisi hasil dari dua kelompok orang (misalnya pria dan wanita) yang memilih 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. Uji tabel chi^2 melakukan hal ini. Hasilnya terlalu besar untuk menolak independensi. Jadi, kami tidak dapat mengatakan, apakah pemungutan suara bergantung 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 mendekati 0, kita simpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

```
  0.0427225484717
```

Beberapa Pengujian Lainnya

Selanjutnya, kami menggunakan analisis varians (uji F) untuk menguji tiga sampel data berdistribusi normal untuk nilai rata-rata yang sama. Metode ini disebut ANOVA (analisis varians). Dalam 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
```

Artinya, kita menolak hipotesis nilai rata-rata yang sama. Kita melakukan ini dengan probabilitas kesalahan sebesar 1,3%.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda dengan menguji median sampel gabungan.

```
>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
```

Uji kesetaraan lainnya adalah uji peringkat. Uji peringkat jauh lebih tajam daripada uji median.

```
>ranktest(a,b)
```

```
  0.00199969612469
```

Dalam contoh berikut, kedua distribusi memiliki rata-rata yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

```
  0.264378760485
```

Sekarang, mari kita coba simulasikan dua perawatan 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];
```

Uji signum memutuskan, apakah a lebih baik dari b.

```
>signtest(a,b)
```

```
  0.0546875
```

Ini adalah kesalahan yang sangat besar. Kita tidak dapat menolak bahwa a sama baiknya dengan b.
Uji Wilcoxon lebih tajam daripada uji ini, tetapi bergantung pada nilai kuantitatif perbedaannya.

```
>wilcoxon(a,b)
```

```
  0.0296680599405
```

Mari kita coba dua pengujian lagi menggunakan seri yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

```
  0.00359458419569
```

```
>wilcoxon(normal(1,20),normal(1,20))
```

```
  0.910521943401
```

Angka Acak

Berikut ini adalah pengujian untuk generator angka acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan masalah apa pun.
Pertama-tama kita menghasilkan sepuluh juta angka acak dalam [0,1].

```
>n:=10000000; r:=random(1,n);
```

Berikutnya kita hitung jarak antara dua angka kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Terakhir, kami memplot berapa kali setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
>  plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```

Hapus data.

```
>remvalue n;
```

Pendahuluan bagi Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Akan tetapi, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi, EMT dapat memenuhi kebutuhan dasar. Lagi pula, EMT dilengkapi dengan paket numerik dan sistem aljabar komputer.

Buku catatan ini ditujukan bagi Anda yang sudah familier dengan R, tetapi perlu mengetahui perbedaan sintaksis EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami melihat cara untuk bertukar data antara kedua sistem.

Harap dicatat bahwa ini adalah pekerjaan yang masih dalam tahap pengerjaan.

Sintaksis Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Dalam EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Selain itu, operator ini memiliki daya pengikatan 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 guna menggabungkan berbagai hal.

Contoh berikut ini, seperti banyak contoh lainnya, berasal dari "Interoduction to R" yang disertakan dalam proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti alurnya 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 digantikan oleh fungsi seq() di R. Kita dapat menulis fungsi ini dalam 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 dalam 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 penugasan. Operator "->" digunakan untuk unit dalam EMT.

```
>125km -> " miles"
```

```
  77.6713990297 miles
```

Operator "<-" untuk penugasan menyesatkan dan bukan ide yang baik untuk R. Berikut ini akan membandingkan a dan -4 dalam EMT.

```
>a=2; a<-4
```

```
  0
```

Dalam R, "a<-4<3" berfungsi, tetapi "a<-4<-3" tidak. Saya juga mengalami ambiguitas serupa dalam EMT, tetapi mencoba menghilangkannya sedikit demi sedikit.

EMT dan R memiliki vektor bertipe boolean. Namun dalam EMT, angka 0 dan 1 digunakan untuk mewakili false dan true. Dalam R, nilai true dan false tetap dapat digunakan dalam aritmatika biasa seperti dalam EMT.

```
>x<5, %*x
```

```
  [0,  0,  1,  0,  0]
  [0,  0,  3.1,  0,  0]
```

EMT memunculkan 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, string dapat berupa string Unicode. String unicode dapat diterjemahkan ke pengodean 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 pada sistem Anda sebagai A dengan titik dan garis di atasnya. Hal ini bergantung pada font yang Anda gunakan.

```
>chartoutf([480])
```

Penggabungan string dilakukan dengan "+" atau "|". String dapat menyertakan angka, yang akan dicetak dalam format saat ini.

```
>"pi = "+pi
```

```
  pi = 3.14159265359
```

Pengindeksan

Sering kali, ini akan berfungsi seperti di R.

Namun EMT akan menginterpretasikan indeks negatif dari belakang vektor, sementara 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 logika tidak diperlakukan secara berbeda sebagai indeks dalam EMT, berbeda dengan R. Anda perlu mengekstrak elemen bukan nol terlebih dahulu dalam 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]
```

Namun, nama untuk indeks tidak dimungkinkan dalam EMT. Untuk paket statistik, hal 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)]; ... ...
               ^

  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 terdapat vektor yang terus bertambah. Anda dapat menetapkan vektor numerik kosong v dan menetapkan nilai ke elemen v[17]. Hal ini tidak mungkin dilakukan 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 yang ditambahkan pada tumpukan dan menyalin vektor itu kembali ke variabel global v.

Yang lebih efisien mendefinisikan vektor terlebih dahulu.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Yang lebih efisien mendefinisikan vektor terlebih dahulu.

```
>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 penggabungan string sederhana. Namun, 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 keluaran.

```
>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 terdapat 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, kita ingin menghitung rata-rata pendapatan di wilayah tersebut. Sebagai program statistik, R memiliki factor() dan tappy() untuk ini.
EMT dapat melakukan ini 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 tersebut, kita dapat menulis fungsi loop kita sendiri untuk melakukan sesuatu hanya untuk satu faktor.
Atau kita dapat 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
```

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 dapat mencetak pendapatan negara rata-rata di wilayah dengan cara yang mudah.

```
>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 disimpan dalam suatu koleksi dengan jenis dan kategori (negara bagian dan teritori dalam contoh kita). Untuk EMT, kita tambahkan 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 dapat meniru tapply() dengan cara berikut. Fungsi 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 pemeriksaan tipe di sini. Satu-satunya tindakan pencegahan menyangkut kategori (faktor) tanpa data. Namun, seseorang 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 data ini 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. Vektor 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 di R.

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

```
          2          3
          7          8
         12         13
         17         18
```

Namun, dalam R dimungkinkan untuk menetapkan daftar indeks vektor tertentu ke suatu nilai. Hal yang sama dimungkinkan dalam 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 dalam EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalinnya dalam fungsi tersebut.

```
>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
```

Produk luar dalam EMT hanya dapat dilakukan antara vektor. Hal ini dilakukan secara otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan yang lainnya berupa 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 terdapat sebuah contoh, yang menghitung distribusi ab-cd untuk a,b,c,d yang dipilih secara acak dari 0 hingga n. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan table() di atasnya.

Tentu saja, ini dapat dicapai dengan loop. Namun, loop tidak efektif dalam EMT atau R. Dalam EMT, kita dapat menulis loop dalam C dan itu akan menjadi solusi tercepat.

Namun, 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"):
```

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 termudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

```
>plot2d(q,distribution=11):
```

Namun, Anda juga dapat menghitung terlebih dahulu jumlah dalam interval yang dipilih. 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="/"):
```

Daftar

EMT memiliki dua jenis daftar. Satu 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 dalam EMT. Ia berperilaku seperti struktur dalam C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

```
>L={{"Fred","Flintstone",40,[1990,1992]}}
```

```
  Fred
  Flintstone
  40
  [1990,  1992]
```

Saat ini unsur-unsur tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Unsur-unsur tersebut diakses dengan angka.

```
>(L[4])[2]
```

```
  1992
```

Input dan Output File (Membaca dan Menulis Data)

Anda sering kali ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang berbagai cara untuk mencapainya. Fungsi sederhana adalah writematrix() dan readmatrix().

Mari kita tunjukkan cara membaca dan menulis vektor bilangan real ke dalam file.

```
>a=random(1,100); mean(a), dev(a),
```

```
  0.4789
  0.27521
```

Untuk menulis data ke dalam berkas, kami menggunakan fungsi writematrix().

Karena pengantar ini kemungkinan besar berada di dalam direktori, tempat pengguna tidak memiliki akses tulis, kami menulis data ke direktori beranda pengguna. Untuk buku catatan sendiri, ini tidak diperlukan, karena berkas data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita tulis vektor kolom a' ke dalam berkas. Ini menghasilkan satu angka di setiap baris berkas.

```
>writematrix(a',filename);
```

Untuk membaca data, kita menggunakan readmatrix().

```
>a=readmatrix(filename)';
```

Dan hapus berkasnya.

```
>fileremove(filename);
>mean(a), dev(a),
```

```
  0.4789
  0.27521
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem bahasa Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (nilai default dipisahkan dengan koma). File berikut "test.csv" akan muncul di folder Anda saat ini.

```
>filename="test.csv"; ...
>writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka berkas ini langsung dengan Excel Indonesia.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut.

```
>s1:="f m m f m m m f f f m m f";  ...
>s2:="f f f m m f f";
```

Untuk menokenisasi ini, kami mendefinisikan vektor token.

```
>tok:=["f","m"]
```

```
  f
  m
```

Lalu 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 tajuk 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();
```

Berkasnya tampak seperti ini.

```
>printfile(file)
```

```
  A,B,C
  0.6210691015193581,0.1861287498944177,0.1334118765499487
  0.5425891336612759,0.8541459276352096,0.5265435039793132
  0.8286381971250506,0.9319592738601091,0.9548321883942634
```

Fungsi readtable() dalam bentuk yang paling sederhana dapat membacanya dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Koleksi ini dapat dicetak dengan writetable() ke buku catatan, atau ke berkas.

```
>writetable(L,wc=10,dc=5)
```

```
         A         B         C
   0.62107   0.18613   0.13341
   0.54259   0.85415   0.52654
   0.82864   0.93196   0.95483
```

Matriks nilai adalah elemen pertama L. Perhatikan bahwa mean() dalam EMT menghitung nilai rata-rata baris matriks.

```
>mean(L[1])
```

```
   0.31354
   0.64109
   0.90514
```

Berkas CSV

Pertama, mari kita tulis matriks ke dalam berkas. Untuk output, kita buat berkas di direktori kerja saat ini.

```
>file="test.csv";  ...
>M=random(3,3); writematrix(M,file);
```

Berikut ini isi berkas tersebut.

```
>printfile(file)
```

```
0.5509346722762319,0.9591222272838416,0.1515283397770994
0.7091161391115151,0.5510213984347968,0.163059576815024
0.3482813265192847,0.6028612820632797,0.4552416781408445
```

CVS ini dapat dibuka pada sistem bahasa Inggris ke Excel dengan mengklik dua kali. Jika Anda mendapatkan berkas tersebut pada sistem bahasa Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Namun, titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari berkas dengan readmatrix().

```
>readmatrix(file)
```

```
   0.55093   0.95912   0.15153
   0.70912   0.55102   0.16306
   0.34828   0.60286   0.45524
```

Dimungkinkan untuk menulis beberapa matriks ke dalam satu berkas. Perintah open() dapat membuka berkas untuk ditulis dengan parameter "w". Nilai default untuk membaca adalah "r".

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh baris kosong. Untuk membaca matriks, buka berkas 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 lembar kerja 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 ini contohnya.

```
>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 mengubahnya di pengaturan sistem atau di Excel, tetapi tidak diperlukan untuk membaca matriks ke EMT.

Cara termudah untuk membaca ini ke 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]'):
```

Ada cara yang lebih mendasar untuk membaca data dari sebuah berkas. Anda dapat membuka berkas dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari sebaris data. Secara default, fungsi ini mengharapkan titik desimal. Namun, fungsi ini juga dapat menggunakan koma desimal, jika Anda memanggil setdecimaldot(",") sebelum menggunakan fungsi ini.

Fungsi berikut adalah contohnya. Fungsi ini akan berhenti di akhir berkas 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)
```

```
    0.55093         0    0.95912         0    0.15153
    0.70912         0    0.55102         0    0.16306
    0.34828         0    0.60286         0    0.45524
```

Semua angka dalam berkas itu juga dapat dibaca dengan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
    0.55093         0    0.95912
          0   0.15153    0.70912
          0   0.55102          0
```

Jadi sangat mudah untuk menyimpan vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

```
  0.49382
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
  0.49382
```

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Misalnya, kita menulis tabel dengan tajuk baris dan kolom ke dalam sebuah berkas.

```
>file="test.tab"; M=random(3,3);  ...
>open(file,"w");  ...
>writetable(M,separator=",",labc=["one","two","three"]);  ...
>close(); ...
>printfile(file)
```

```
  one,two,three
        0.04,       0.02,       0.96
        0.54,       0.09,       0.87
        0.13,       0.41,        0.9
```

Ini dapat diimpor ke Excel.

Untuk membaca berkas di EMT, kami menggunakan readtable().

```
>{M,headings}=readtable(file,>clabs); ...
>writetable(M,labc=headings)
```

```
         one        two      three
        0.04       0.02       0.96
        0.54       0.09       0.87
        0.13       0.41        0.9
```

Menganalisis Garis

Anda bahkan dapat mengevaluasi setiap garis secara manual. 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 membuat token pada baris tersebut.

```
>vt=strtokens(line)
```

```
  2020-11-03
  Tue
  1’114.05
```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang tepat.

```
>day(vt[1]),  ...
>indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])),  ...
>strrepl(vt[3],"’","")()
```

```
  7.3816e+05
  2
  1114
```

Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data.

Asumsikan kita memiliki baris berikut sebagai 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 mengekstraknya, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung tutup >,
- string apa pun yang tidak mengandung tanda kurung dengan sub-kecocokan "(...)",
- tanda kurung buka dan tutup menggunakan solusi terpendek,
- lagi-lagi string apa pun yang tidak mengandung tanda kurung,
- dan tanda kurung buka <.

Ekspresi reguler agak sulit dipelajari tetapi sangat ampuh.

```
>{pos,s,vt}=strxfind(line,">([^<>]+)<.+?>([^<>]+)<");
```

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-kecocokan.

```
>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 berkas dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh ini, kami membaca versi terkini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." dalam 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 2024-01-12
```

Input dan Output Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke dalam 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 = [ ..
  0.4623879917445496, 0.3180836434648883;
  0.3529402068815742, 0.7817441427373571;
```

Sekarang kita dapat memuat berkas tersebut. Berkas tersebut akan mendefinisikan matriks M.

```
>load(file); show M,
```

```
  M =
    0.46239   0.31808
    0.35294   0.78174
```

Ngomong-ngomong, jika writevar() digunakan pada suatu variabel, ia akan mencetak definisi variabel dengan nama variabel ini.

```
>writevar(M); writevar(inch$)
```

```
  M = [ ..
  0.4623879917445496, 0.3180836434648883;
  0.3529402068815742, 0.781744142737357];
  inch$ = 0.0254;
```

Kita juga dapat membuka berkas baru atau menambahkannya ke berkas yang sudah ada. Dalam contoh ini, kita menambahkannya ke berkas yang dibuat sebelumnya.

```
>open(file,"a"); ...
>writevar(random(2,2),"M1"); ...
>writevar(random(3,1),"M2"); ...
>close();
>load(file); show M1; show M2;
```

```
  M1 =
    0.44086   0.37666
    0.12362   0.26942
  M2 =
    0.87284
    0.40248
    0.78955
```

Untuk menghapus file apa pun gunakan fileremove().

```
>fileremove(file);
```

Vektor baris dalam sebuah berkas tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita buat berkas seperti itu, tulis 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 = [
  0.234994356974
  0.183264664178
  0.755219384652
  0.602458087359
  0.543445468039
  ];
```

```
>load(file); M
```

```
  [0.23499,  0.18326,  0.75522,  0.60246,  0.54345]
```

catatan : ketika mengenter perintah-perintah diatas ternyata hasil yang didapatkan berbeda-beda

Nomor 1 Carilah rata-rata dan standar deviasi beserta plot dari data berikut
X = 2000,2500,2700,3500,4500,5000

```
>X=[2000,2500,2700,3500,4500,5000]; ...
>mean(X), dev(X),
```

```
  3366.7
  1186
```

```
>aspect(1.5); boxplot(X):
```

Nomor 2
Misalkan diberikan data skor hasil statistika dari 20 orang mahasiswa sebagai berikut:
70,65,79,90,60,79,86,95,100,70,60,91,68,84,59,90,88,84,86,90

Tentukan rata-rata dari data tersebut!

```
>X=[70,65,79,90,60,79,86,95,100,70,60,91,68,84,59,90,88,84,86,90]
```

```
  [70,  65,  79,  90,  60,  79,  86,  95,  100,  70,  60,  91,  68,  84,
  59,  90,  88,  84,  86,  90]
```

```
>mean(X)
```

```
  79.7
```

```
>
```

```
>
```