TUGAS PEKAN 16 APLIKASI KOMPUTER

BAB LaTeX dan Markdown

Chintya Wijayanti
22305144029
Matematika E 2022

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

DAFTAR ISI

BAB 1

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

[a4paper,10pt]article eumat

EMT untuk Perhitungan Aljabar

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

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

Contoh pertama

Menyederhanakan bentuk aljabar:

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

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

Menjabarkan:

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

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

Baris Perintah

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

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

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

```
16.7551608191
```

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

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

```
50.2654824574
100.530964915
```

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

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

```
0.540302305868
```

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

```
0.857553215846
```

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

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

```
1.41666666667
1.41421568627
1.41421356237
```

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

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

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

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

```
81
```

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

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

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

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

```
1.41421356237
```

Struktur bersyarat juga berfungsi.

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

```
Thought so!
```

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

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

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

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

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

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

```
2.5
```

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

Sintaks Dasar

Euler tahu fungsi matematika biasa. Seperti yang Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilainya, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut kuadrat dalam Euler. Tentu saja, x^(1/2) juga dimungkinkan.

Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Tapi ruang antara perintah diharapkan.

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

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

```
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Memasuki

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

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

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

```
8.77908249441
```

Untuk menghitung ekspresi rumit seperti

lateks: \left(\frac{\frac17 + \frac18 + 2}{\frac13 + \frac12}\right)^2 \pi

Anda harus memasukkannya dalam bentuk baris.

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

```
23.2671801626
```

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

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

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

```
0.47619047619
10/21
```

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

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

```
14.4978445072
```

Operator numerik Euler meliputi

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

dan masih banyak lagi.

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

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

Beberapa perintah memiliki alias, mis. Untuk log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

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

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

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

Bilangan Asli

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

```
>longest 1/3
```

```
      0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

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

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

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

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

String

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

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

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

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

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

```
  The area of the circle with radius 2 cm is 12.5663706144 cm^2.
```

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

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

```
Golden Ratio : 1.61803
```

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

```
>none
```

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

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

```
1234.5
```

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

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

```
affe
charlie
bravo
```

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

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

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

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

String Unicode dapat digabungkan seperti string lainnya.

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

```
 = 45°
```

I

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

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

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

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

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

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

```
Ü is a German letter
```

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

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

```
We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

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

```
Ähnliches
```

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

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

```
0
1
```

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

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

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

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

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

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

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

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

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

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

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

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
      3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
    0.33333
    3.14159
    0.84147
```

Standarnya adalah format (12).

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

```
0.333333333333
```

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

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

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

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

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

```
  3.1416
```

Fungsi "format terpanjang" mengatur format skalar juga.

```
>longestformat; pi
```

```
  3.141592653589793
```

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

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

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

Tetapi format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
  3.141592653589793
```

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

```
    3.14159
```

The default is defformat().

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

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

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

```
        4.934802200544679
```

Ada juga operator pendek untuk mencetak hasil dalam format pecahan. Kami sudah 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 dengan tepat. Kesalahan bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

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

```
   -1.110223024625157e-16
```

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

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

```
  0
```

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

Variabel global dapat digunakan dalam evaluasi.

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

```
        12.56637061435917
```

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

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

```
  -0.919535764538
```

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

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

```
  36
```

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

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

```
  45
```

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

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

```
45
```

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

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

```
12
```

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

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

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

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

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

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

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

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

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

```
-0.475
```

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

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

```
-0.425
```

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

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

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

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

```
>$&44!
```

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

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

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

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

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

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

Anda akan belajar bahwa yang berikut ini juga berfungsi.

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

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

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

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

Dengan begitu Anda dapat menggunakan solusi persamaan dalam persamaan lain.

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

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

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

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

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

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


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

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

Sekali lagi, % mengacu pada hasil sebelumnya.

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

```
2432902008176640000
```

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

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

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

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

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

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

```
 2
g
```

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

```
 3  x
x  E
```

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

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

```
                                        5
                                   125 E


                             18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

Untuk evaluasi ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator "with".

Baris perintah berikut juga menunjukkan bahwa Maxima dapat mengevaluasi ekspresi secara numerik dengan float().

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

```
                                   10         5
                             1000 E   - 125 E


                            2.20079141499189e+7
```

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

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

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

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

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

Penugasan juga bisa bersifat simbolis.

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

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

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

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

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

```
1.56155281281
```

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

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

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

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

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

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

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

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

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

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

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

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

```
>f(2)
```

```
4.472135955
```

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

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

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

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

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

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

```
0.786151377757
```

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

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

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

```
0.707106781187
```

Lebih baik kita menghapus redefinisi dosa ini.

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

```
0.707106781187
```

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

```
16
```

Menyetelnya akan menimpa nilai default.

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

```
80
```

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

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

```
16
```

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

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

```
24
```

Tetapi parameter yang ditetapkan menimpa nilai global.

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

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

```
20
```

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

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

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

```
0.703467422498
```

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

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

```
0.703467422498
```

```
>function P(x,n) &= (2*x-1)^n; $&P(x,n)
>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 &= fungsinya simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya.

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

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

lateks: f(x) = \int_1^x t^t \, dt,

yang tidak dapat dinilai secara simbolis.

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

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

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

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.

Tetapi fungsi tersebut juga dapat digunakan untuk vektor numerik.

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

```
17
```

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

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

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

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

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

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

Untuk meringkas

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

Ekspresi dapat diselesaikan secara numerik dan simbolis.

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

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

```
1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

```
>$&solve(x^2=2,x)
>$&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 dapat diubah dengan variabel yang ditetapkan.
Kami menggunakan y=2 dan memeriksa dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
0.966715594851
2
```

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

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

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

```
>longest sol()
```

```
-0.6180339887498949       1.618033988749895
```

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

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

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

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

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

lateks: aˆx-xˆa = 0.1

dengan a=3.

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

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

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

```
2.54116291558
```

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

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

```
2.54116291558
```

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])
```

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

```
              1              2
              3              4
```

Produk matriks dilambangkan dengan titik.

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

```
              3
              4
```

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

```
  [3,  4]
```

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

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

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

```
             11
             25
```

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

```
              1              0
              0              1
```

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

```
>A.A
```

```
             7            10
            15            22
```

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

```
             1             4
             9            16
```

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

```
            37            54
            81           118
```

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

```
            37            54
            81           118
```

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

```
             1             1
             1             1
```

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

```
      0.333333      0.666667
          0.75             1
```

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

```
            -2
           2.5
```

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

```
            -2
           2.5
```

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

```
             1             0
             0             1
```

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

```
             1             0
             0             1
```

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

```
             1             4
             9            16
```

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

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

```
             9
            16
```

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

```
>2*A
```

```
             2             4
             6             8
```

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

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

```
            1             4
            3             8
```

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

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

```
            2             6
            6            12
```

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

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

```
            1             2
            1             2
```

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

```
            1             4
            3             8
```

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

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

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

Sekali lagi, ingat bahwa ini bukan produk matriks!

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

```
55
```

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

```
55
```

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

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

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

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

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

```
5
```

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

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

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

Dari vektor seperti itu, "bukan nol" memilih elemen bukan nol.
Dalam hal ini, kami mendapatkan indeks semua elemen lebih besar dari 50.

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

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

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

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

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

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

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

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

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

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

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

```
112
```

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

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

Indeks ini dapat digunakan untuk mengatur elemen ke beberapa nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

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

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

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

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

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

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

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

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

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

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

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

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

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

```
             1             1
             2             4
             3             1
[-0.765761,  -0.952814,  -0.548138]
```

Fungsi Matriks Lainnya (Membangun Matriks)

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

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

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

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

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

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

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

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

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

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

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

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

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

Untuk vektor, ada panjang().

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

```
9
```

Ada banyak fungsi lain, yang menghasilkan matriks.

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

```
              1               1
              1               1
```

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

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

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

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

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

```
       0.66566      0.831835
         0.977      0.544258
```

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

```
>redim(1:9,3,3) // menyusun elemen2 1, 2, 3, ..., 9 ke bentuk matriks 3x3
```

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

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

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

Let us test.

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

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

Fungsi multdup() menduplikasi elemen vektor.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal.

Kita mulai dengan matriks identitas.

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

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

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

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

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

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

Berikut adalah fungsi, yang mengembalikan matriks tri-diagonal.

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

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

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

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

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

Sekarang kita dapat mengekstrak diagonal.

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

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

Misalnya. Kita dapat membagi matriks dengan diagonalnya. Bahasa matriks memperhatikan 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
```

Hampir semua fungsi di Euler juga berfungsi untuk input matriks dan vektor, kapan pun ini masuk akal.

Misalnya, fungsi sqrt() menghitung akar kuadrat dari semua elemen vektor atau matriks.

```
>sqrt(1:3)
```

```
[1,  1.41421,  1.73205]
```

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

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

Dengan ini dan operator titik dua a:delta:b, vektor nilai fungsi dapat dihasilkan dengan mudah.

Pada contoh berikut, kita membangkitkan vektor nilai t[i] dengan spasi 0,1 dari -1 hingga 1. Kemudian kita membangkitkan vektor nilai fungsi

lateks: s = tˆ3-t

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

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

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

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

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

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

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

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

```
55
```

Secara default, vektor baris dicetak dalam format yang ringkas.

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

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

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

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

```
5
25
```

Untuk mentranspos matriks kita menggunakan apostrof.

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

```
            1
            2
            3
            4
```

Jadi kita dapat menghitung matriks A kali vektor b.

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

```
           30
           70
```

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

```
>v'.v
```

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

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

```
>v.v'
```

```
30
```

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

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

```
30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan aturannya.

- Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap elemen.

- Operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan berpasangan ke elemen matriks.

- Jika kedua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.

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

Berikut ini adalah kasus sederhana dengan operator ^.

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

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

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

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

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

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
14
```

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

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

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

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

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

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

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

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

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

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

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

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

```
6
```

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

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

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

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

Dengan menggunakan bahasa matriks, kita dapat menghasilkan beberapa tabel dari beberapa fungsi sekaligus.

Dalam contoh berikut, kita menghitung t[j]ˆi untuk i dari 1 hingga n. Kami mendapatkan matriks, di mana setiap baris adalah tabel tˆi untuk satu i. Yaitu, matriks memiliki elemen lateks: a_{i,j} = t_jˆi, \quad 1 \le j \le 101, \quad 1 \le i \le n

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

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

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

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

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

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

Untuk mengakses elemen matriks, gunakan notasi braket.

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

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

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[2]
```

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

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

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

```
  2
```

Untuk memastikan, Anda mendapatkan baris pertama untuk matriks 1xn dan mxn, tentukan semua kolom menggunakan indeks kedua kosong.

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

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

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

Di sini kita menginginkan baris pertama dan kedua dari A.

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

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

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

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

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

Trik indeks bekerja dengan kolom juga.

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

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

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

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

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

```
                 3
                 6
                 9
```

Atau, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir dari A.

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

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

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

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

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

Kami juga dapat menetapkan nilai ke baris A.

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

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

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

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

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

Selain itu, beberapa jalan pintas diperbolehkan.

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

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

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

```
>A[4]
```

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

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengocok vektor.

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

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

Indeks berisi urutan yang tepat dari v.

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

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

Ini bekerja untuk vektor string juga.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unik mengembalikan daftar elemen unik vektor yang diurutkan.

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

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

Ini bekerja untuk vektor string juga.

```
>unique(s)
```

```
a
aa
d
e
```

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

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

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

```
         -4
        4.5
```

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

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

```
 8.790745908981989e-13
```

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

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

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

Determinan matriks ini adalah 0.

```
>det(A)
```

```
0
```

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

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

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

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

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

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

```
          1          4
          5          2
```

Dalam ekspresi simbolik, gunakan dengan.

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

Akses ke baris matriks simbolik bekerja seperti halnya dengan matriks numerik.

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

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

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

Ada fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk ini, lihat dokumentasi Maxima atau tutorial tentang Maxima di EMT.

```
>v &= makelist(1/(i+j),i,1,3); $v
```

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

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

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

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

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

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

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

```
                         1                         0
                         0                         1
```

Misalnya. nilai eigen dari A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

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

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

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

```
                1       3.14159
                4             5
```

Masih ada perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolis, pendekatan fraksional untuk real akan digunakan.

```
>$&A
```

Untuk menghindarinya, ada fungsi "mxmset(variable)".

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

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

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

Ketepatan angka floating point besar dapat diubah.

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
  4592307816406286208998628034825342117068b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolis apa pun menggunakan "@var".

Perhatikan bahwa ini hanya diperlukan, jika variabel telah didefinisikan dengan ":=" atau "=" sebagai variabel numerik.

```
>B:=[1,pi;3,4]; $&det(@B)
```

Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk perhitungan suku bunga. Kami melakukannya secara numerik dan simbolis untuk menunjukkan kepada Anda bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata.

Asumsikan Anda memiliki modal awal 5000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler akan memahami sintaks berikut juga.

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

```
5150
```

Tetapi lebih mudah menggunakan faktornya

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

```
1.03
5150
```

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

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

```
6719.58189672
```

Untuk tujuan kita, kita dapat mengatur format menjadi 2 digit setelah titik desimal.

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

```
      6719.58
```

Mari kita cetak yang dibulatkan menjadi 2 digit dalam kalimat lengkap.

```
>"Starting from " + K + "$ you get " + round(K*q^10,2) + "$."
```

```
  Starting from 5000$ you get 6719.58$.
```

Bagaimana jika kita ingin mengetahui hasil antara dari tahun 1 sampai tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak harus menulis loop, tetapi cukup masukkan

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

```
  Real 1 x 11 matrix

      5000.00     5150.00     5304.50     5463.64     ...
```

Bagaimana keajaiban ini bekerja? Pertama ekspresi 0:10 mengembalikan vektor bilangan bulat.

```
>short 0:10
```

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

Kemudian semua operator dan fungsi dalam Euler dapat diterapkan pada elemen vektor untuk elemen. Jadi

```
>short q^(0:10)
```

```
  [1,  1.03,  1.0609,  1.0927,  1.1255,  1.1593,  1.1941,  1.2299,
  1.2668,  1.3048,  1.3439]
```

adalah vektor faktor q^0 sampai q^10. Ini dikalikan dengan K, dan kami mendapatkan vektor nilai.

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

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

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

Mari kita bandingkan dua hasil, dengan dan tanpa pembulatan.

```
>longest oneyear(1234.57), longest 1234.57*q
```

```
              1271.61
            1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.

Cara termudah adalah iterasi fungsi, yang mengulangi fungsi tertentu beberapa kali.

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

```
 Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64     ...
```

Kami dapat mencetaknya dengan cara yang ramah, menggunakan format kami dengan tempat desimal tetap.

```
>VKr'
```

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

Untuk mendapatkan elemen tertentu dari vektor, kami menggunakan indeks dalam tanda kurung siku.

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

```
5150.00
5000.00     5150.00     5304.50
```

Anehnya, kita juga bisa menggunakan vektor indeks. Ingat bahwa 1:3 menghasilkan vektor [1,2,3].

Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

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

```
6719.60
6719.58
```

Perbedaannya sangat kecil.

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

```
>function onepay (K) := K*q+R
```

Kita tidak perlu menentukan q atau R untuk definisi fungsi. Hanya jika kita menjalankan perintah, kita harus mendefinisikan nilai-nilai ini. Kami memilih R=200.

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

```
Real 1 x 11 matrix

    5000.00     5350.00     5710.50     6081.82     ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

```
Real 1 x 11 matrix

    5000.00     4950.00     4898.50     4845.45     ...
```

Kami melihat bahwa uang berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama, tetapi menghapus 200, kita kehilangan uang setiap tahun.

Bagaimana kita bisa menentukan berapa tahun uang itu akan bertahan? Kita harus menulis loop untuk ini. Cara termudah adalah dengan iterasi cukup lama.

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

```
Real 1 x 51 matrix

    5000.00     4950.00     4898.50     4845.45     ...
```

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

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

```
48.00
```

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

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

Fungsi iterate() memiliki satu trik lagi. Itu bisa mengambil kondisi akhir sebagai argumen. Kemudian akan mengembalikan nilai dan jumlah iterasi.

```
>{x,n}=iterate("onepay",5000,till="x<0"); x, n,
```

```
-19.83
 47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Apa yang akan menjadi tingkat bunga?

Ini adalah pertanyaan yang hanya bisa dijawab dengan angka. Di bawah ini, kita akan mendapatkan formula yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada formula yang mudah untuk tingkat bunga. Tapi untuk saat ini, kami bertujuan untuk solusi numerik.

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

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

Iterasinya sama seperti di atas

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

Tapi kami tidak lagi menggunakan nilai global R dalam ekspresi kami. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

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

Mari kita coba tes.

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

```
-19.83
```

Sekarang kita bisa menyelesaikan masalah kita.

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

```
3.15
```

Rutin memecahkan memecahkan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami mengambil nilai awal 3% untuk algoritma. Fungsi solve() selalu membutuhkan nilai awal.

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

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

```
-336.08
```

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

Solusi Simbolik untuk Masalah Suku Bunga

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

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

Kita sekarang dapat mengulangi ini.

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

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

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

Ini agak rumit. Jumlahnya dievaluasi dengan bendera "simpsum" untuk menguranginya menjadi hasil bagi.

Mari kita membuat fungsi untuk ini.

```
>function fs(K,R,P,n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K,R,P,n)
```

Fungsi tersebut melakukan hal yang sama seperti fungsi f kita sebelumnya. Tapi itu lebih efektif.

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

```
-19.82504734650985
-19.82504734652684
```

Kita sekarang dapat menggunakannya untuk menanyakan waktu n. Kapan modal kita habis? Dugaan awal kami adalah 30 tahun.

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

```
20.51
```

Jawaban ini mengatakan bahwa itu akan menjadi negatif setelah 21 tahun.

Kita juga dapat menggunakan sisi simbolis Euler untuk menghitung formula pembayaran.

Asumsikan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) meninggalkan sisa hutang sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas

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

Biasanya rumus ini diberikan dalam bentuk

lateks: i = \frac{P}{100}

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

Kita dapat memecahkan tingkat R secara simbolis.

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

Seperti yang Anda lihat dari rumus, fungsi ini mengembalikan kesalahan titik mengambang untuk i=0. Euler tetap merencanakannya.

Tentu saja, kami memiliki batasan berikut.

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

Jelas, tanpa bunga kita harus membayar kembali 10 tarif 500.

Persamaan juga dapat diselesaikan untuk n. Kelihatannya lebih bagus, jika kita menerapkan beberapa penyederhanaan untuk itu.

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

BAB 2

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

[a4paper,10pt]article eumat

NAMA : Chintya Wijayanti
NIM : 22305144029
KELAS : Matematika E

Menggambar Grafik Fungsi Simbolik

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

plot2d() menerima ekspresi, fungsi, dan data.

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

Keterangan:(menggambar grafik fungsi satu variabel yang fungsinya didefinisikan sebagai fungsi simbolik)
- &: untuk menampilkan variabel pada teks

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

```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
>plot2d("f",0,1;0.4): // plot with a=0.4
>plot2d({{"f",0.2}},0,1); // plot with a=0.2
>plot2d({{"f(x,b)",b=0.1}},0,1): // plot with 0.1
>function f(x) := x^3-x;...
>plot2d("f",r=1):
```

Berikut merupakan ringkasan dari fungsi yang diterima
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f

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

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

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

```
>plot2d(f,0,2):
>expr &= sin(x)*exp(-x)
```

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

```
>plot2d(expr,0,3pi):
>function f(x) &=x^x;
>plot2d(f,r=1,cx=1,cy=1,color=red,thickness=2)
>plot2d(&diff(f(x),x),>add,color=blue,style="-.-"):
```

Untuk gaya garis ada berbagai pilihan.
- gaya=”...”. Pilih dari ”-”,”–”,”-.”,”.”,”.-.”,”-.-”.
- warna: Lihat di bawah untuk warna.
- ketebalan: Default adalah 1.

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

```
>plot2d("exp(-x^2)",r=2,color=blue,thickness=3,style="--"):
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15):
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```

Menggambar Beberapa Kurva Sekaligus

Dalam subtopik ini, kita akan membahas mengenai cara menggambar beberapa kurva sekaligus. Dalam hal ini kita dapat menggambar beberapa kurva dalam jendela grafik yang berbeda secara bersama-sama. Untuk membuat ini kita dapat menggunakan perintah figure(). Berikut contoh dari menggambar beberapa kurva sekaligus

Menggambar plot fungsi

$$x^n, 1 \leq n \leq 4$$

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

Penjelasan sintaks dari plot fungsi

$$x^n, 1 \leq n \leq 4$$

- reset;
Perintah ini berguna untuk menghapus grafik yang telah ada sebelumnya, sehingga kita dapat memulai dari awal untuk menggambar grafik
- figure(2x2);
Perintah figure() digunakan untuk membuat jendela grafik dengan ukuran
axb. Dalam kasus ini perintah figure(2,2) memiliki makna bahwa jendela grafik yang dibuat berukuran 2x2. Artinya, akan ada empat jendela grafik yang akan ditampilkan dengan tata letak 2 baris dan 2 kolom.
- for n=1 to 4;
Perintah ini digunakan untuk melakukan pengulangan (looping) perintah sebanyak empat kali, yaitu dari 1 hingga 4.
- figure(n);
Perintah ini digunakan untuk beralih dari jendela grafik satu ke jendela grafik lainnya (jendela grafik ke-n).
- plot2d("x^"+n);
Perintah plot2d() digunakan untuk membuat plot fungsi matematika.
Dalam hal ini fungsi yang diplot adalah x^n, di mana n adalah nilai dari variabel yang sedang diulang. Dengan kata lain, ini akan membuat
plot dari x^1, x^2, x^3, dan x^4 dalam jendela grafik yang sesuai
- end;
Perintah ini menandakan akhir dari looping.
- figure(0);
Perintah ini digunakan untuk beralih kembali ke jendela grafik utama.

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

Dari sini dapat kita perhatikan untuk membuat kurva fungsi xˆn (x pangkat n) perintahnya tidak ditulis dengan (xˆn) melainkan ditulis dengan ("xˆ"+n). Tanda petik dua ("...") digunakan untuk mengidentifikasi bahwa teks tersebut merupakan ekspresi matematika.
Sedangkan tanda (+) digunakan untuk menggabungkan string dengan nilai yang berubah-ubah atau variabel.

Contoh lain:
Menggambar plot fungsi

$$f(x) = x^3 - x, -2 < x < 2$$

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

```
Penjelasan sintaks dari plot fungsi
```

$$f(x) = x^3 - x, -2 < x < 2$$

- reset;
Perintah ini berguna untuk menghapus grafik yang telah ada sebelumnya, sehingga kita dapat memulai dari awal untuk menggambar grafik
- figure (3,3);
Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x3. Artinya, akan ada empat jendela grafik yang akan ditampilkan dengan tata letak 3 baris dan 3 kolom.
- for k=1:9;
Perintah ini digunakan untuk melakukan pengulangan (looping) perintah sebanyak sembilan kali.
- figure(n);
Perintah ini digunakan untuk beralih dari jendela grafik satu ke

```
jendela grafik lainnya (jendela grafik ke-n).
```

- plot2d("xˆ3-x",-2,2,grid=k);
Perintah plot2d() digunakan untuk membuat plot fungsi matematika.
Dalam hal ini fungsi yang diplot adalah xˆ3-x, dengan batas sumbu x dari -2 hingga 2. Argumen grid=k digunakan untuk mengaktifkan grid pada jendela grafik ke-k.
- end;
Perintah ini menandakan akhir dari looping.
- figure(0);
Perintah ini digunakan untuk beralih kembali ke jendela grafik utama.

Dari contoh diatas dapat kita perhatikan bahwa tampilan plot dari yang ke-1 hingga ke-9 memiliki tampilan yang berbeda-beda. Dalam EMT memiliki berbagai gaya plot 2D yang dapat dijalankan menggunakan perintah grid=n dimana n adalah jumlah langkah minimal. Setiap nilai n memiliki tampilan plot adaptif yang berbeda dalam plot 2D, diantaranya yaitu:
0 : tidak ada grid (kisi), frame, sumbu, dan label, hanya kurva saja
1 : dengan sumbu, label-label sumbu di luar frame jendela grafik
2 : tampilan default
3 : dengan grid pada sumbu x dan y, label-label sumbu berada di dalam jendela grafik
4 : tidak ada grid (kisi), sumbu x dan y, dan label berada di luar frame jendela grafik
5 : tampilan default tanpa margin di sekitar plot

6 : hanya dengan sumbu x y dan label, tanpa grid
7 : hanya dengan sumbu x y dan tanda-tanda pada sumbu.
8 : hanya dengan sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.
9 : tampilan default dengan tanda-tanda kecil di dalam jendela
10: hanya dengan sumbu-sumbu, tanpa tanda

Contoh lain:
Menggambar plot fungsi

$$g(x) = 2x^3 - x$$

```
>reset;
>aspect(1.2);
>figure(3,4); ...
> figure(2); plot2d("2x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("2x^3-x",grid=2); ... // default ticks
>figure(4); plot2d("2x^3-x",grid=3); ... // x-y- axis with labels inside
>figure(5); plot2d("2x^3-x",grid=4); ... // no ticks, only labels
>figure(6); plot2d("2x^3-x",grid=5); ... // default, but no margin
>figure(7); plot2d("2x^3-x",grid=6); ... // axes only
>figure(8); plot2d("2x^3-x",grid=7); ... // axes only, ticks at axis
>figure(9); plot2d("2x^3-x",grid=8); ... // axes only, finer ticks at axis
>figure(10); plot2d("2x^3-x",grid=9); ... // default, small ticks inside
>figure(11); plot2d("2x^3-x",grid=10); ...// no ticks, axes only
>figure(0):
```

Penjelasan sintaks dari plot fungsi

$$g(x) = 2x^3 - x$$

- aspect(1.2);
Perintah aspect() digunakan untuk mengatur rasio aspek dari jendela grafik. Hal ini berarti perintah aspect(1.2); akan menghasilkan plot dengan perbandingan rasio panjang dan lebar 2:1.
- figure(3,4);
Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x4.
Jadi, akan ada total 12 jendela grafik yang akan ditampilkan dalam tata letak 3 baris dan 4 kolom.
- figure(1); plot2d("x^3-x",grid=0); ...
Adalah perintah untuk beralih ke jendela grafik pertama dan menggambar plot dari fungsi x^3 - x tanpa grid, frame, atau sumbu.
- figure(2); plot2d("x^3-x",grid=1); ...
Adalah perintah untuk beralih ke jendela grafik kedua dan menggambar plot dari fungsi x^3 - x dengan grid hanya pada sumbu x dan y.
- figure(3); plot2d("x^3-x",grid=2); ...
Adalah perintah untuk beralih ke jendela grafik ketiga dan menggambar plot dari fungsi x^3 - x dengan tampilan default, termasuk tanda-tanda default pada sumbu.
- figure(4); plot2d("x^3-x",grid=3); ...
Adalah perintah untuk beralih ke jendela grafik keempat dan menggambar plot dari fungsi x^3 - x dengan grid pada sumbu x dan y, serta label-label sumbu yang ada di dalam jendela.
- figure(5); plot2d("x^3-x",grid=4); ...
Adalah perintah untuk beralih ke jendela grafik kelima dan menggambar plot dari fungsi x^3 - x tanpa tanda-tanda sumbu, hanya label-label yang ada.
- figure(6); plot2d("x^3-x",grid=5); ...
Adalah perintah untuk beralih ke jendela grafik keenam dan menggambar plot dari fungsi x^3 - x dengan

tampilan default, tetapi tanpa margin di sekitar plot.
- figure(7); plot2d(”xˆ3-x”,grid=6); ...
Adalah perintah untuk beralih ke jendela grafik ketujuh dan menggambar plot dari fungsi xˆ3 - x hanya dengan sumbu-sumbu (tanpa grid atau label).
- figure(8); plot2d(”xˆ3-x”,grid=7); ...
Adalah perintah untuk beralih ke jendela grafik kedelapan dan menggambar plot dari fungsi xˆ3 - x hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu.
- figure(9); plot2d(”xˆ3-x”,grid=8); ...
Adalah perintah untuk beralih ke jendela grafik kesembilan dan menggambar plot dari fungsi xˆ3 - x hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.
- figure(10); plot2d(”xˆ3-x”,grid=9); ...
Adalah perintah untuk beralih ke jendela grafik kesepuluh dan menggambar plot dari fungsi xˆ3 - x dengan tanda-tanda default kecil di dalam jendela.
- figure(11); plot2d(”xˆ3-x”,grid=10); ...
Adalah perintah untuk beralih ke jendela grafik kesebelas dan menggambar plot dari fungsi xˆ3 - x hanya dengan sumbu-sumbu, tanpa tanda-tanda.
- figure(0);
Adalah perintah untuk beralih kembali ke jendela grafik utama atau jendela grafik dengan nomor 0 setelah semua perintah dalam urutan selesai dieksekusi.

Dari ketiga contoh di atas, dapat kita katakan bahwa untuk menggambar beberapa kurva sekaligus itu dapat dilakukan dengan satu baris perintah ataupun dengan cara mendefinisikannya 1 per 1.

Terlihat beberapa jenis grid memiliki tampilan yang mirip atau sama, seperti 1 dan 2, 2 dan 5, 4 dan 9, 7 dan 8, untuk dapat membedakannya secara lebih jelas, ubah grid dari contoh di bawah ini.

```
>reset;
>aspect(1.3);
>figure(1,3);...
>figure (1); plot2d("x^2*exp(-x)",0,10);...
>figure (2); plot2d("2*exp(x)",-5,5);...
>figure (3); plot2d("exp(x^2)",-2,2);...
>figure (0):
```

Contoh lain:

```
>reset;
>aspect(3/4);
>figure(2,1);...
>for a=1:2; figure(a); plot2d("2*x*log(x^2)",0,3,grid=a); end;...
>figure(0):
```

* pada bidang koordinat yang sama

Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, kecuali panggilan pertama.

Berikut contohnya:
menggambar kurva

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

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

```
>aspect(); plot2d("cos(x)",r=3); plot2d("x^2",style=".",>add):
```

$$f(x) = cos(x) - 1$$

$$f(x) = sin(x) - 1$$

```
>aspect(2); plot2d("cos(x)-1",-1,6); plot2d("sin(x)-1",style="--",>add):
```

Selain menggunakan >add kita juga bisa menambahkannya secara langsung

Berikut contohnya:
Menggambar kurva

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

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

```
>plot2d(["2x+1","x"],0,8):
```

$$f(x) = sin(2x)$$

$$f(x) = cos(3x)$$

```
>aspect(1.5); plot2d(["sin(2x)","cos(3x)"],0,8):
```

Kegunaan >add yang lain juga bisa untuk menambahkan titik pada kurva.

Berikut contohnya:
Menambahkan sebuah titik di

$$f(x) = x + 4$$

```
>aspect(); plot2d("x+4",-2,5,); plot2d(2,6,>points,>add):
```

Kita juga bisa mencari titik perpotongan dengan cara berikut:

$$sin(x) = 2x$$

```
>plot2d(["sin(x)","2x"],r=2,cx=1,cy=1, ...
>  color=[black,blue],style=["-","."], ...
>  grid=1);
>x0=solve("sin(x)-2x",1);  ...
>  plot2d(x0,x0,>points,>add);  ...
>  label("sin(x) = 2x",x0,x0,pos="cl",offset=20):
>function f(x,a) := x^2+a*x-x/a; ...
>plot2d("f",-10,10;1,title="a=1"):
> plot2d({{"f",1}},-10,10); ...
>for a=1:10; plot2d({{"f",a}},>add); end:
>function f(x,a) := x^2*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
>plot2d({{"f",1}},-8,8); ...
>for a=2:5; plot2d({{"f",a}},>add,thickness=2); end:
>aspect(2.1); &plot2d(1/x,[x,-1,1]):
>x=linspace(-1,1,50);...
>plot2d("1/x"):
```

BAB 3

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

[a4paper,10pt]article eumat

IDENTITAS

Nama : Chintya Wijayanti
NIM : 22305144029
Kelas : Matematika E 2022

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

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

Euler dapat merencanakan

- permukaan dengan bayangan dan garis level atau rentang level,
- awan poin,
- kurva parametrik,
- permukaan implisit.

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

```
>aspect(1.5); plot3d("x^2+sin(y)",r=pi):
```

Untuk grafik fungsi, gunakan

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

Standarnya adalah kotak kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, tetapi plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Ini bisa diubah.

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

Kami menggunakan default n=40 dan grid=10.

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

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

- kiri, kanan, atas, bawah: putar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih memutar sumber cahaya (lihat di bawah)
- spasi: reset ke default
- kembali: akhiri 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 (defaultnya adalah <fscale).
skala: angka atau vektor 1x2 untuk skala ke arah x dan y.
bingkai: jenis bingkai (default 1).

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

Tampilan dapat diubah dengan berbagai cara.

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

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

```
>view
```

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

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

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=2,angle=0,height=0):
```

Plot terlihat selalu ke pusat kubus plot. Anda dapat memindahkan pusat dengan parameter tengah.

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

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

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

Jika Anda mematikannya dengan scale=false, Anda harus berhati-hati, agar plot tetap pas dengan jendela plot, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengahnya.

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

```
Closing bracket missing in function call!
Error in:
... ^2-y^2)",r=2,<fscale,<scale,distance=13,height=50?,   center=[ ...
                                                        ^
```

Sebuah plot kutub juga tersedia. Parameter polar=true menggambar plot polar. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```

Rotasi parameter memutar fungsi dalam x di sekitar sumbu x.

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

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

Berikut adalah plot dengan tiga fungsi.

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

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

-> hue: Menyalakan bayangan cahaya alih-alih kabel.
-> kontur: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Sebuah vektor nilai untuk garis kontur.

Standarnya 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 grid yang lebih halus untuk 100x100 poin, skala 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?):
```

```
Closing bracket missing in function call!
Error in:
... ,  >contour,>spectral,fscale=1,scale=1.1,angle=45?,height=20?) ...
                                                    ^
```

```
>plot3d("exp(x*y)",angle=100?,>contour,color=green):
```

```
Closing bracket missing in function call!
Error in:
plot3d("exp(x*y)",angle=100?,>contour,color=green): ...
                           ^
```

Bayangan default menggunakan warna abu-abu. Tetapi rentang warna spektral juga tersedia.

-> spektral: 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 alih-alih rentang level.

```
>plot3d("x^2-y^2",0,1,0,1,angle=220?,level=-1:0.2:1,color=redgreen):
```

```
Closing bracket missing in function call!
Error in:
plot3d("x^2-y^2",0,1,0,1,angle=220?,level=-1:0.2:1,color=redgr ...
                                  ^
```

Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.

Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

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

```
Closing bracket missing in function call!
Error in:
... 2+y^3",level=[-0.1,0.9;0,1],   >spectral,angle=30?,grid=10,con ...
                                                 ^
```

Dalam contoh berikut, kami memplot himpunan, di mana

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

Kami menggunakan satu garis tipis untuk garis level.

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

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

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

Berikut adalah beberapa gaya lagi. Kami selalu mematikan frame, dan menggunakan berbagai skema warna untuk plot dan grid.

```
>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 lainnya, bernomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan warna=nilai, di mana nilai

- spektral: untuk rentang dari biru ke merah
- putih: untuk rentang yang lebih redup
-kuningbiru,ungu hijau,birukuning,hijaumerah
- 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. Itu juga dapat diatur dengan parameter.

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

Perhatikan bahwa program tidak membuat perbedaan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda perlu 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):
```

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

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

dapat divisualisasikan dalam potongan sejajar dengan bidang x-y-, x-z- dan y-z.

- implisit=1: potong sejajar dengan bidang y-z
- implisit=2: potong sejajar dengan bidang x-z
- implisit=4: potong sejajar dengan bidang x-y

Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh kita plot

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

```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x-, y- dan z, atau tiga fungsi atau ekspresi fx(x,y), fy(x,y), fz(x,y).

$$\gamma(t,s) = (x(t,s), y(t,s), z(t,s))$$

Karena x,y,z adalah matriks, kita asumsikan bahwa (t,s) melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

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

```
Need , or ; between matrix elements in strict mode
Error in:
... grid=[10,20], values=s,contourcolor=red,level=[90?-24?;90?-22? ...
                                                   ^
```

Berikut adalah contoh, yang merupakan grafik fungsi.

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

Namun, kita bisa membuat segala macam permukaan. Berikut adalah permukaan yang sama dengan fungsi

$$x = y\,z$$

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

```
Closing bracket missing in function call!
Error in:
plot3d(t*s,t,s,angle=180?,grid=10): ...
                         ^
```

Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan bayangan dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

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

dengan

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

Kami mendistorsi ini dengan sebuah faktor

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

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

Tentu saja, titik cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik-titik tersebut.

Gayanya sama seperti di plot2d dengan points=true;

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

Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung titik-titik kurva. Untuk kurva di bidang kami menggunakan urutan koordinat dan parameter wire=true.

```
>t=linspace(0,8pi,500); ...
>function f(x)
```

```
  endfunction
```

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

EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata merah/sian.

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

```
  Closing bracket missing in function call!
  Error in:
   plot3d("x^2+y^3",>anaglyph,>contour,angle=30?): ...
                                               ^
```

Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian 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 dalam ruang.

Untuk demo berikut, kami mengatur parameter u- dan v-, dan menghasilkan koordinat ruang dari ini.

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

Berikut adalah contoh yang lebih rumit, yang megah dengan kacamata merah/sian.

```
>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 bar 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 bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.

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

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

Dimungkinkan untuk membagi plot permukaan menjadi dua atau lebih bagian.

```
>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 menghasilkan matriks data M dari 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 individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40?,zoom=1.8):
```

```
Closing bracket missing in function call!
Error in:
... 0*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]):
```

```
>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
```

Kami ingin memutar kurva hati di sekitar sumbu y. Berikut adalah ungkapan, yang mendefinisikan hati:

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

Selanjutnya kita atur

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

```
>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 hati 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?):
```

```
Closing bracket missing in function call!
Error in:
... ,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.

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

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

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

```
Closing bracket missing in function call!
Error in:
... ymax=1.2,zmin=-1.2,zmax=1.4, implicit=1,angle=-30?,zoom=2.5,n= ...
                                                     ^
```

Fungsi plot3d bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, dimungkinkan untuk mendapatkan plot berbingkai dari objek apa pun yang Anda suka.

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

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

```
    y=0:0.01:1; x=(0.1:0.01:1)';
    plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
      hues=0.5,>contour,color=orange);
    h=holding(1);
    plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
    holding(h);
  endfunction
```

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[0.1,1,0,1,0,1],angle=-45?, ...
>  center=[0,0,-0.7],zoom=6):
```

```
  Closing bracket missing in function call!
  Error in:
  framedplot("myplot",[0.1,1,0,1,0,1],angle=-45?,   center=[0,0, ...
                                                ^
```

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

```
>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 frame untuk menghitung animasi terlebih dahulu.

Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi memanggil addpage() untuk setiap plot baru. Akhirnya itu menjiwai plot.

Silakan pelajari sumber rotasi untuk melihat lebih detail.

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

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

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

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

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

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

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

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

```
>load povray;
```

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

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

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

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

Jika Anda memulai 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 dijalankan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengakui dialog awal Povray.

```
>pov3d("x^2+y^2",zoom=3);
```

Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kami juga dapat menambahkan garis level ke plot fungsi.

```
>pov3d("x^2+y^3",axiscolor=red,angle=20?, ...
>  look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot himpunan titik di bidang kompleks, di mana produk dari jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=1.5, ...
>  angle=-120?,level=1/40,dlevel=0.005,light=[-1,1,1],height=45?,n=50, ...
>  <fscale,zoom=3.8);
```

Alih-alih fungsi, kita dapat memplot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Dalam contoh kita memutar fungsi di sekitar sumbu z.

```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,8)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40?,height=20?,axis=0,zoom=4,light=[10,-5,5]);
```

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

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

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3;  ...
>pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...
>  w=500,h=300);
```

Dengan metode bayangan canggih dari Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayang-bayang triknya mungkin menjadi jelas.

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

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

```
 2  3
x  y
```

Persamaan permukaannya adalah [x,y,Z]. Kami menghitung dua turunan ke x dan y ini dan mengambil produk silang sebagai normal.

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

Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

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

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

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 dilakukan oleh A. Busser di Povray. Ada versi yang ditingkatkan dari ini dalam contoh.

see: Contoh\Trefoil Simpul | 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, ketiga fungsi koordinat sebagai ekspresi simbolik.

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

Kemudian kedua vektor turunan ke x dan y.

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

Sekarang normal, yang merupakan produk silang dari dua turunan.

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

Kami sekarang mengevaluasi semua ini secara numerik.

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

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

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

Kami juga dapat menghasilkan 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();
```

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

Kami memulai output dengan povstart().

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

Pertama kita 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(green)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c1
```

```
cylinder { <-1,0,0>, <1,0,0>, 1
 texture { pigment { color rgb <0.564706,0.0627451,0.0627451> }  }
 finish { ambient 0.2 }
 }
```

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

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

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

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

Sekarang kita mendefinisikan objek persimpangan, dan menulis hasilnya ke file.

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

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

```
>povend;
```

Fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi povbox() mengembalikan string, 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 memotong satu objek dari yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

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

Untuk demonstrasi ini, kami mendefinisikan objek di Povray, alih-alih menggunakan string di Euler. Definisi ditulis ke file segera.

Koordinat kotak -1 berarti [-1,-1,-1].

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

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

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

Kami menghasilkan kubus kedua, dan memutar dan menskalakannya sedikit.

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

Kemudian 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, seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsinya sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

```
>povstart(angle=70?,height=50?,zoom=4);
```

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

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

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

Kami ingin memaksimalkan xy di bawah kondisi x+y=1 dan menunjukkan sentuhan tangensial dari garis level.

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

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

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

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

Sekarang kita mendefinisikan 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);
```

Tulis permukaan dikurangi dua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tulis dua perpotongan.

```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulis titik maksimum.

```
>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();
```

Untuk menghasilkan anaglyph untuk kacamata merah/sian, Povray harus berjalan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kacamata merah/sian untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki sakelar sederhana untuk menghasilkan anaglyphs.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45?,>anaglyph, ...
>  center=[0,0,0.5],zoom=3.5);
```

Jika Anda membuat adegan dengan objek, Anda perlu menempatkan generasi adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
```

```
    s=povsphere(povc,1);
    cl=povcylinder(-povz,povz,0.5);
    clx=povobject(cl,rotate=xrotate(90?));
    cly=povobject(cl,rotate=yrotate(90?));
    c=povbox([-1,-1,0],1);
    un=povunion([cl,clx,cly,c]);
    obj=povdifference(s,un,povlook(red));
    writeln(obj);
    writeAxes();
  endfunction
```

Fungsi povanaglyph() melakukan semua ini. Parameternya seperti di povstart() dan povend() digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```

Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Tapi Anda tidak terbatas pada 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 kami.

```
>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 menempatkan objek-objek ini ke dalam sebuah adegan. Untuk tampilan, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0?,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

>povend();

memanggil program Povray. Namun, jika terjadi kesalahan, itu tidak menampilkan kesalahan. Karena itu Anda harus menggunakan

```
  >povend(<keluar);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```

Berikut adalah contoh yang lebih rumit. Kami memecahkan

$$Ax \leq b, \quad x \geq 0, \quad c.x \to \text{Max.}$$

dan menunjukkan titik layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi sama sekali.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0,  1,  0.5]
```

Ya, sudah.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah bidang

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
  endfunction
```

Kemudian kita mendefinisikan perpotongan semua setengah ruang 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 bisa 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 yang optimal.

```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>  povlook(red,0.9)));
```

Dan kesalahan dalam arah optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kita tambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan mengubahnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=125?)); ...
>povend();
```

Lebih Banyak Contoh

Anda dapat menemukan beberapa contoh lagi untuk Povray di Euler di file berikut.

See: Examples/Dandelin Spheres
See: Examples/Donat Math
See: Examples/Trefoil Knot
See: Examples/Optimization by Affine Scaling

BAB 4

KB PEKAN 6-7: MENGGUNAKAN EMT UNTUK KALKULUS

[a4paper,10pt]article eumat

Nama : Chintya Wijayanti
NIM : 22305144029
Kelas: Matematika E 2022

Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format nama_fungsi := rumus fungsi (untuk 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.

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
Variable or function a not found.
Error in:
f(a) // tidak dapat dihitung nilainya ...
   ^
```

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

```
0
```

```
>g(0)
```

```
0
```

```
>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^2-3*x)/(x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
    ^
```

Silakan Anda plot kurva fungsi di atas

```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))
```

```
2.20920171961
```

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 koorinat.

```
>f(0:10) // nilai-nilai f(1), f(2), ..., f(10)
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

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

```
  if x>0 then return x^3
  else return x^2
  endif;
endfunction
```

```
>f(1)
```

```
1
```

```
>f(-2)
```

```
4
```

```
>f(-5:5)
```

```
[25,  16,  9,  4,  1,  0,  1,  8,  27,  64,  125]
```

```
>aspect(1.5); plot2d("f(x)",-5,5):
>function f(x) &= 2*E^x // fungsi simbolik
```

```
                                 x
                              2 E
```

```
>$f(a) // nilai fungsi secara simbolik
>f(E)
```

```
30.308524483
```

```
>$f(E)
>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):
```

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-sungsi tersebut dan komposisinya 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 fungsi-fungsi tersebut dan komposisi - komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Nomor 1

$$a(x) = x^3 - 4x^2 + 5x - 6$$

```
>function a(x) &= (x^3+4*x^2+5*x-6) // fungsi simbolik
```

```
                    3      2
                   x  + 4 x  + 5 x - 6
```

```
>function a(x) := (x^3+4*x^2+5*x-6) // fungsi numerik
>a(5)
```

```
  244
```

```
>a(-5:5)
```

```
  [-56,  -26,  -12,  -8,  -8,  -6,  4,  28,  72,  142,  244]
```

```
>aspect(2); plot2d("a(x)",-5,5):
```

Nomor 2

$$g(x) = \sqrt{x^2 + 4}$$

```
>function g(x) := (sqrt(x^2+4)) // fungsi numerik
>g(2)
```

```
  2.82842712475
```

```
>g=(-2:2)
```

```
[-2,  -1,  0,  1,  2]
```

```
>aspect(2); plot2d("g(x)",-2,2):
```

Nomor 3

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

```
>function d(x) := ((x^3+2)/(3*x)) // fungsi numerik
>d(4)
```

```
5.5
```

```
>d=(-4:4)
```

```
[-4,  -3,  -2,  -1,  0,  1,  2,  3,  4]
```

```
>aspect(2); plot2d("d(x)",-4,4):
```

Nomor 4

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

$$g(x) = \sin x$$

```
>function f(x) &= (cos(x)) // fungsi numerik
```

```
cos(x)
```

```
>f(pi)
```

-1

```
>f(2*pi)
```

1

```
>function g(x) &= (sin(x)) // fungsi numerik
```

sin(x)

```
>g(pi)
```

0

```
>g(2*pi)
```

0

```
>f(g(pi)) // komposisi fungsi
```

1

```
>g(f(pi))
```

-0.841470984808

```
>function h(x) &= f(g(pi))
```

f(g(pi))

```
>plot2d("h(x)",-2,2):
```

Nomor 5

$$e(x) = \begin{cases} -x^2+4 & x \leq 1 \\ 3x & x > 1. \end{cases}$$

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

```
if x<=1 then return -x^2+4
else return 3*x
endif;
endfunction
```

```
>e(5)
```

```
15
```

```
>e=(-5:5)
```

```
[-5,  -4,  -3,  -2,  -1,  0,  1,  2,  3,  4,  5]
```

```
>aspect(2); plot2d("e(x)",-5,5):
```

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und' (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

maxima: 'limit((xˆ3-13*xˆ2+51*x-63)/(xˆ3-4*xˆ2-3*x+18),x,3)=limit((xˆ3-13*xˆ2+51*x-63)/(xˆ3-4*xˆ2-3*x+18),x,3)

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4); plot2d(3,-4/5,>points,style="ow",>
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

maxima: 'limit(2*x*sin(x)/(1-cos(x)),x,0)=limit(2*x*sin(x)/(1-cos(x)),x,0)

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add):
>$limit(cot(7*h)/cot(5*h),h,0)
```

maxima: showev('limit(cot(7*h)/cot(5*h),h,0))

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
>aspect(1.5); plot2d("((x/8)^(1/3)-1)/(x-8)",-1,1); plot2d(8,1/24,>points,style="ow",>add):
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(1/(2*x-1),x,0))
>plot2d("1/(2*x-1)",-2,2); plot2d(0,-1,>points,style="ow",>add):
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
>plot2d("(x^2-3*x-10)/(x-5)",-8,8); plot2d(5,7,>points,style="ow",>add):
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
>plot2d("(sqrt(x^2+x)-x)",-1,1); plot2d(0,1/2,>points,style="ow",>add):
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(sin(x)/x,x,0))
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
>$showev('limit(sin(x^3)/x,x,0))
>plot2d("sin(x^3)/x",-pi,pi); plot2d(0,0,>points,style="ow",>add):
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(log(x), x, minf))
>$showev('limit((-2)^x,x, inf))
>$showev('limit(t-sqrt(2-t),t,2,minus))
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
>plot2d("x-sqrt(2-x)",0,2):
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
>plot2d("(x^2-9)/(2*x^2-5*x-3)",-1,1); plot2d(3,6/7,>points,style="ow",>add):
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((1-cos(x))/x,x,0))
>plot2d("(1-cos(x))/x",-pi,pi); plot2d(0,0,>points,style="ow",>add):
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
>plot2d("(x^2+abs(x))",-pi,pi); plot2d(0,-1,>points,style="ow",>add):
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((1+1/x)^x,x,inf))
>plot2d("(1+1/x)^x",0,1000):
>$showev('limit((1+k/x)^x,x,inf))
>$showev('limit((1+x)^(1/x),x,0))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((x/(x+k))^x,x,inf))
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(sin(1/x),x,0))
>$showev('limit(sin(1/x),x,inf))
>plot2d("sin(1/x)",-5,5):
```

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

Nomor 1

```
>$showev('limit((x^2+2*x-1),x,2))
>$showev('limit((x^2+2*x-1),x,4))
>$showev('limit((x^2+2*x-1),x,inf))
>plot2d("(x^2+2*x-1)",-8,8); plot2d(2,7,>points,style="ow",>add):
```

Nomor 2

```
>$showev('limit((x^4 + 2*x^3 - x^2)/(x^2),x,0))
>$showev('limit((x^4 + 2*x^3 - x^2)/(x^2),x,1))
>$showev('limit((x^4 + 2*x^3 - x^2)/(x^2),x,inf))
>plot2d("(x^4 + 2*x^3 - x^2)",-1,1); plot2d(0,-1,>points,style="ow",>add):
```

Nomor 3

```
>$showev('limit(sqrt(3*x-5),x,inf))
>$showev('limit(sqrt(3*x-5),x,0))
>$showev('limit(sqrt(3*x-5),x,2))
>plot2d("sqrt(3*x-5)",-5,5); plot2d(2,1,>points,style="ow",>add):
```

Nomor 4

```
>$showev('limit(((x^2)+(2*x)),x,0,plus))
>$showev('limit(((x^2)+(2*x)),x,2,minus))
>$showev('limit(((x^2)+(2*x)),x,inf))
>plot2d("((x^2)+(2*x))",0,8):
```

Nomor 5

```
>$showev('limit((1-cos(x))/x^2,x,0))
>$showev('limit((1-cos(x))/x^2,x,inf))
>$showev('limit((1-cos(x))/x^2,x,2))
>plot2d("(1-cos(x))/x^2",-5,5):
```

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)^n-x^n)/h,h,0)) // turunan x^n
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan (x+h)^n dengan menggunakan teorema binomial.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Untuk

$$f(x) = x^n$$

maka

$$f'(x) = \lim_{h \to 0} \frac{(x+h)^n - x^n}{h}$$
$$f'(x) = \lim_{h \to 0} \frac{(x^n + \frac{n}{1!}x^{n-1}h + \frac{n(n-1)}{2!}x^{n-2}h^2 + \frac{n(n-1)(n-2)}{3!}x^{n-3}h^3 + ...) - x^n}{h}$$
$$f'(x) = \lim_{h \to 0} \frac{n.x^{n-1}h + \frac{n(n-1)}{2!}x^{n-2}h^2 + \frac{n(n-1)(n-2)}{3!}x^{n-3}h^3 + ...}{h}$$
$$f'(x) = \lim_{h \to 0} n.x^{n-1} + \frac{n(n-1)}{2!}.x^{n-2}h + \frac{n(n-1)(n-2)}{3!}.x^{n-3}h^2 + ...$$
$$f'(x) = n.x^{n-1} + 0 + 0 + ... + 0$$
$$f'(x) = n.x^{n-1}$$

Jadi, terbukti bahwa

$$\lim_{h\to 0}\frac{(x+h)^n-x^n}{h}=n.x^{n-1}$$

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

Untuk

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

maka

$$f'(x)=\lim_{h\to 0}\frac{sin(x+h)-sin(x)}{h}$$
$$f'(x)=\lim_{h\to 0}\frac{sin(x)cos(h)+cos(x)sin(h)-sin(x)}{h}$$
$$f'(x)=\lim_{h\to 0}\frac{sin(x)(cos(h)-1)+cos(x)sin(h)}{h}$$
$$f'(x)=\lim_{h\to 0}sin(x).\frac{cos(h)-1}{h}+\lim_{h\to 0}cos(x).\frac{sin(h)}{h}$$
$$f'(x)=sin(x).0+cos(x).1$$
$$f'(x)=cos(x)$$

Jadi, terbukti bahwa

$$\lim_{h\to 0}\frac{sin(x+h)-sin(x)}{h}=cos(x)$$

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

untuk

$$f(x) = log(x)$$

maka

$$f'(x) = \lim_{h \to 0} \frac{log(x+h) - log(x)}{h}$$
$$f'(x) = \lim_{h \to 0} \frac{log\frac{x+h}{x}}{h}$$
$$f'(x) = \lim_{h \to 0} log(\frac{x+h}{x})^{\frac{1}{h}}$$
$$f'(x) = \lim_{h \to 0} log(1 + \frac{h}{x})^{\frac{1}{h}}$$
$$f'(x) = log \lim_{h \to 0} (1 + \frac{h}{x})^{\frac{1}{h}}$$
$$f'(x) = loge^{\frac{1}{x}}$$
$$f'(x) = \frac{1}{x} loge$$
$$f'(x) = \frac{1}{x}$$

Jadi, terbukti bahwa

$$\lim_{h \to 0} \frac{log(x+h) - log(x)}{h} = \frac{1}{x}$$

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
>$showev('limit((E^(x+h)-E^x)/h,h,0))// turunan f(x)=e^x
```

```
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

Use assume!
Error in:
                                         ^
```

Maxima bermasalah dengan limit:

$$\lim_{h\to 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>.,$showev('limit((E^h-1)/h,h,0))
>$showev('factor(E^(x+h)-E^x))
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
>function f(x) &= x^x
```

```
                                      x
                                     x
```

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h\to 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

```
[x > 0]
```

```
>&forget(x<0)
```

```
[x < 0]
```

```
>&facts()
```

```
[]
```

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

```
sinh(x)
```

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

Hasilnya adalah cosh(x), karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

```
>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?

```
>$showev('diff(f(x),x))
>$% with x=3
```

```
Maxima said:
diff: variable must not be a number; found: 3
#0: with(expr='diff(('diff(f(x),x,1))(x),x,1) = 'diff(('diff(f(x),x,1))(x),x,1),eq=x = 3)
 -- an error. To debug this try: debugmode(true);

Error in:
           ^
```

```
>$float(%)
>plot2d(f,0,3.1):
```

```
Function f needs at least one argument!
Use: map f (x)
Error in:
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
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

Perhatikan titik-titik "puncak" grafik y=f(x) dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), seperti contoh-contoh tersebut. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

Nomor 1

```
>function f(x) := cos(x^2)
>$showev('limit((cos((x+h)^2)- cos(x^2))/h,h,0))
>plot2d(["f(x)","df(x)"], -pi, pi, color=[green,red]):
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

Nomor 2

```
>function f(x) := sqrt(x^2+6)
>$showev('limit((sqrt((x+h)^2+6)-sqrt(x^2+6))/h,h,0))
>plot2d(["f(x)","df(x)"],-pi,pi,color=[green,red]):
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

Nomor 3

```
>function f(x) :=(2-x)^4
>$showev('limit(((2-(x+h))^4-(2-x)^4)/h,h,0))
>plot2d(["f(x)","df(x)"],-pi,pi,color=[green,red]):
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

Nomor 4

```
>function f(x) :=2*sin(x)+3*cos(x)
>$showev('limit((2*sin(x+h)+3*cos(x+h)-(2*sin(x)+3*cos(x)))/h,h,0))
>plot2d(["f(x)","df(x)"],-pi,pi,color=[green,red]):
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

Nomor 5

```
>function f(x) :=5*x-4
>$showev('limit(((5*(x+h)-4)-(5*x-4))/h,h,0))
>plot2d(["f(x)","df(x)"],-pi,pi,color=[green,red]):
```

```
Syntax error in expression, or unfinished expression!
df:
    useglobal; return 'limit((f(x+h)-f(x))/h,h,0)
Error in expression: df(x)
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x)\ dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x))
```

```
Answering "Is n equal to -1?" with "no"
```

```
>$showev('integrate(1/(1+x),x))
>$showev('integrate(1/(1+x^2),x))
>$showev('integrate(1/sqrt(1-x^2),x))
>$showev('integrate(sin(x),x,0,pi))
>$showev('integrate(sin(x),x,a,b))
>$showev('integrate(x^n,x,a,b))
```

```
Answering "Is n positive, negative or zero?" with "positive"
```

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
>$ratsimp(%)
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
>$factor(%)
>function map f(x) &= E^(-x^2)
```

```
                                        2
                                     - x
                                    E
```

```
>$showev('integrate(f(x),x))
```

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$erf(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}}\, dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

maxima: 'integrate(f(x),x,0,pi)

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```

Integral tentu

maxima: 'integrate(f(x),x,0,pi)

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva y=f(x) tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
>// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

maxima: 'integrate(f(x),x,0,pi) = 0.1*sum(fx[i],i,1,length(fx))

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai f(x) untuk x = 0.1, 0.2, 0.3, ..., 3.2.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

```
12.4
```

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

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

```
                                     x
                                    x
```

```
>$showev('integrate(f(x),x,0,1))
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

```
                                 2     5
                              sin (3 x  + 7)
```

```
>integrate(f,0,1)
```

```
 0.542581176074
```

```
>&showev('integrate(f(x),x,0,1))
```

```
         1                           1              pi
        /                     gamma(-) sin(14) sin(--)
        [     2    5                 5              10
        I  sin (3 x  + 7) dx = ------------------------
        ]                                   1/5
        /                                10 6
         0
       4/5                  1          4/5                  1
 - (((6    gamma_incomplete(-, 6 I) + 6    gamma_incomplete(-, - 6 I))
                            5                               5
             4/5                    1
 sin(14) + (6    I gamma_incomplete(-, 6 I)
                                    5
    4/5                    1                       pi
 - 6    I gamma_incomplete(-, - 6 I)) cos(14)) sin(--) - 60)/120
                           5                       10
```

```
>&float(%)
```

```
         1.0
        /
        [       2      5
        I    sin (3.0 x  + 7.0) dx =
        ]
        /
         0.0
0.09820784258795788 - 0.008333333333333333
 (0.3090169943749474 (0.1367372182078336
 (4.192962712629476 I gamma__incomplete(0.2, 6.0 I)
 - 4.192962712629476 I gamma__incomplete(0.2, - 6.0 I))
 + 0.9906073556948704 (4.192962712629476 gamma__incomplete(0.2, 6.0 I)
 + 4.192962712629476 gamma__incomplete(0.2, - 6.0 I))) - 60.0)
```

```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

```
>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); ...
>plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add): // Plot daerah antara 2 kurva
>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
```

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

```
0.0758191713542
```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
>$float(%)
```

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

$$r(t)=1+\frac{\sin(3t)}{2},\quad 0\leq t\leq 2\pi.$$

```
>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): // Kita gambar kurvanya terlebih dahulu
>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( ...
                                                     ^
```

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
```

```
9.0749467823
```

Spiral Logaritmik

$$x=e^{ax}\cos x,\ y=e^{ax}\sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
>&kill(a) // hapus expresi a
```

```
done
```

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

```
Maxima said:
fullmap: arguments must have same formal structure.
 -- an error. To debug this try: debugmode(true);

Error in:
function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t) ...
                                   ^
```

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

```
Maxima said:
fullmap: arguments must have same formal structure.
 -- an error. To debug this try: debugmode(true);

Error in:
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 // panjang kurva (spiral)
```

```
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 // panjang kurva (spiral) ...
                               ^
```

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

```
8.78817491636
```

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
>$float(%)
>x=-1:0.2:1; y=x^2; plot2d(x,y);  ...
>  plot2d(x,y,points=1,style="o#",add=1):
```

Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

```
2.95191957027
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Koordinat Kartesius

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
>plot2d(z,r=2,level=0,n=100):
```

Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```

Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%,x); $sol
```

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
```

Fungsi tersebut juga dapat digunaka untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l*x, yakni x=f(l) dan y=l*f(l).

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```

Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)^2+diff(l*f(l),l)^2)); $'ds(l)=ds(l)
>$integrate(ds(l),l,0,1)
```

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral etrsebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

```
4.91748872168
```

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

```
4.91748872168
```

Perhitungan di datas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...
```

```
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
return romberg(ds,a,b);
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya
```

```
2.95788571509
```

Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, bandingkan hasilnya!
```

```
4.91748872168
```

Kita hitung panjang spiral Archimides berikut ini dengan fungsi tersebut.

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

```
21.2562941482
```

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

```
done
```

```
>function ds(fx,fy) &&= sqrt(diff(fx,x)^2+diff(fy,x)^2)
```

```
                  2               2
         sqrt(diff (fy, x) + diff (fx, x))
```

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
```

```
Maxima said:
Too many arguments supplied to ds(l); found: [x*cos(x),x*sin(x)]
 -- an error. To debug this try: debugmode(true);

Error in:
sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk mengh ...
                            ^
```

```
>$sol | trigreduce | expand, $integrate(%,x,0,2*pi), %()
```

```
21.2562941482
```

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
```

```
Maxima said:
Too many arguments supplied to ds(l); found: [3*x^2-1,3*x^3-1]
 -- an error. To debug this try: debugmode(true);

Error in:
sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol ...
                                  ^
```

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%) // panjang kurva di atas
```

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r. Posisi titik pusat lingkaran pada saat t adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula (0,0) dan posisinya pada saat t adalah:

$$(r(t - \sin(t)), r(1 - \cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika t=0, t=pi/2, t=r*pi.

```
>x &= r*(t-sin(t))
```

```
                              r (t - sin(t))
```

```
>y &= r*(1-cos(t))
```

```
                              r (1 - cos(t))
```

Berikut kita gambar sikloid untuk r=1.

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
>  plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2,ex(2)],[1,ey(2)],color=red,>add); ...
>  plot2d(ex(2),ey(2),>points,>add,color=red); ...
>  plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2pi,ex(2pi)],[1,ey(2pi)],color=red,>add);  ...
>  plot2d(ex(2pi),ey(2pi),>points,>add,color=red):
```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen panjang kurva sikloid
>ds &= trigsimp(ds); $ds
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

```
8
```

```
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
```

```
8
```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

$$2\pi.$$

Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2pi sejauh 2piR.)
Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = (x(s),\ y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$\mathbf{T}(s) = (x'(s),\ y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka T'(s) ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\begin{aligned}\mathbf{T}(s) &= \gamma'(s),\\ \mathbf{T}^2(s) &= 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s)\cdot\mathbf{T}(s) = 0\\ \kappa(s) &= \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.\end{aligned}$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm\kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:
Akan ditentukan kurvatur lingkaran

$$x = r\cos t,\ y = r\sin t.$$

```
>fx &= r*cos(t); fy &=r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen panjang kurva, pan
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... =integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen ...
                                                     ^
```

```
>&kill(s); fx &= r*cos(s/r); fy &=r*sin(s/r); // definisi ulang persamaan parametrik terhadap s dengan
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan menggunakan
```

Untuk representasi parametrik umum, misalkan

$$x = x(t),\ \ y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned}\kappa = \frac{d\phi}{ds} &= \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}.\end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2\phi \frac{d\phi}{dt} = \frac{d}{dt}(\tan\phi) = \frac{d}{dt}\left(\frac{dy}{dx}\right) = \frac{d}{dt}\left(\frac{dy/dt}{dx/dt}\right) = \frac{d}{dt}\left(\frac{y'(t)}{x'(t)}\right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}.$$

$$\begin{aligned}\frac{d\phi}{dt} &= \frac{1}{\sec^2\phi}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1+\tan^2\phi}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1+\left(\frac{y'(t)}{x'(t)}\right)^2}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}.\end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t),\ \ y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta)\cos\theta,\ \ y = r(\theta)\sin\theta,$$

maka rumus kurvaturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{\left(r'(\theta)^2 + r'(\theta)^2\right)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:
Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r\cos t,\ y = r\sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t,\ y = f(t),\quad \text{dengan } x'(t) = 1,\ x''(t) = 0,$$

sehingga kurvaturnya adalah

$$\kappa(t) = \frac{y''(t)}{\left(1 + y'(t)^2\right)^{3/2}}.$$

Contoh:
Akan ditentukan kurvatur parabola

$$y = ax^2 + bx + c.$$

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4),\ 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah (-1/2, 5/4).

```
>plot2d(["f(x)", "k(x)"],-2,1, color=[blue,red]); plot2d("1/k(x)",-1.5,1,color=green,>add); ...
>plot2d("-1/2+1/k(-1/2)*cos(x)","5/4+1/k(-1/2)*sin(x)",xmin=0,xmax=2pi,>add,color=blue):
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x, y) = 0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x},\ F_y = \frac{\partial F}{\partial y},\ F_{xy} = \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial x}\right),\ F_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial x}\right),\ F_{yy} = \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial y}\right),$$

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } \frac{dy}{dx} = -\frac{F_x}{F_y},$$

sehingga kurvaturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{\left(F_x^2 + F_y^2\right)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:
Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0.$$

```
>function F(x,y) &=a*x^2+b*x+c-y; $F(x,y)
>Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y),y), Fxy &=diff(diff(F(x,y),x),y), Fyy &
```

```
2 a x + b

  2 a

  - 1

   0

   0
```

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2); $'k(x)=k(x) // kurvatur parabola t
```

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
- Gunakan integral tentu untuk menghitung volume benda putar kurva y= f(x) yang diputar mengelilingi sumbu x dari x=a sampai x=b, yakni

$$V=\int_a^b \pi(f(x)^2\ dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)
- Gunakan integral tentu untuk menghitung panjang kurva y=f(x) dari x=a sampai x=b dengan menggunakan rumus:

$$S=\int_a^b \sqrt{1+(f'(x))^2}\ dx.$$

(Pilih fungsi dan gambar kurvanya.)
- Apabila fungsi dinyatakan dalam koordinat kutub x=f(r,t), y=g(r,t), r=h(t), x=a bersesuaian dengan t=t0 dan x=b bersesuian dengan t=t1, maka rumus di atas akan menjadi:

$$S=\int_{t_0}^{t_1} \sqrt{x'(t)^2+y'(t)^2}\ dt.$$

- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk:

```
a. koordinat Kartesius (persamaan y=f(x))
b. koordinat kutub ( r=r(theta))
c. persamaan parametrik x=x(t), y=y(t)
d. persamaan implit F(x,y)=0
```

- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

Nomor 1

```
>function f(x):=sin(3x)
>$showev('integrate(sin(2*x),x))
>$showev('integrate(sin(2*x),x,0,pi/2))
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Nomor 2

```
>function f(x):=sqrt(2*x^3+3*x)
>$showev('integrate(f(x),x))
>$showev('integrate(sqrt(2*x^3+3*x),x,0,2))
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Nomor 3

```
>function f(x):=(x^5+x)
>$showev('integrate((x^5+x),x))
>$showev('integrate(x^5+5,x,0,3))
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Nomor 4

```
>function f(x):=cos(2x)
>$showev('integrate(cos(2*x),x))
>$showev('integrate(cos(2*x),x,0,pi/2))
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Nomor 5

```
>function f(x):=5*x^3+4*x+2
>$showev('integrate(5*x^3+4*x+2,x))
>$showev('integrate(5*x^3+4*x+2,x,0,pi/2))
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Nomor 6

```
>function f(x):=cos(x)
>$showev('integrate(cos(x),x))
>$showev('integrate(cos(x),x,0,pi/2))
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh pengguanaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1,  3,  5,  7,  9,  11,  13,  15,  17,  19,  21,  23,  25,  27,  29]
```

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

$$x_{k+1} = g(x_k),\ x_0 = x_0, k = 1, 2, 3, ..., n.$$

Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

```
>q=1.05; iterate("x*q",1000,n=10)'
```

```
   1000
   1050
 1102.5
1157.63
1215.51
1276.28
 1340.1
 1407.1
1477.46
1551.33
1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```

Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar r dan biaya administrasi a, pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)
```

```
[1000,  994,  987.98,  981.93,  975.86,  969.76,  963.64]
```

```
>iterate({{"saldo",0.005,10}},2000,n=6)
```

```
[2000,  1998,  1995.99,  1993.97,  1991.95,  1989.92,  1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
```

```
[2500,  2500,  2500,  2500,  2500,  2500,  2500]
```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)
```

```
[3000,  3002,  3004.01,  3006.03,  3008.05,  3010.08,  3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()". Fungsi ini menghitung nilai-nilai x[n] dari semua nilai sebelumnya, x[1],...,x[n-1] yang diketahui.
Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, \quad x_1 = 1, \quad x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
```

```
[1,  1,  2,  3,  5,  8,  13,  21,  34,  55,  89,  144,  233,  377,  610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$'(1+sqrt(5))/2=float((1+sqrt(5))/2)
>plot2d(sequence("x[n-1]+x[n-2]",[1,1],250)^(1/(1:250))):
```

Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah (n-1) elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah 2ˆ(n-2), untuk n=2, 4, 5, ...

```
>sequence("sum(x)",1,10)
```

```
[1,  1,  2,  4,  8,  16,  32,  64,  128,  256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

$$x_n = A \cdot x_{n-1},$$

dengan A suatu matriks 2x2, dan setiap x[n] merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],6)
```

```
Real 2 x 6 matrix

            1             2             5            13     ...
            1             3             8            21     ...
```

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak log_2(n).

$$x_n = A.x_{n-1} = A^2.x_{n-2} = A^3.x_{n-3} = ... = A^{n-1}.x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,6)
```

```
Real 2 x 6 matrix

            1             5            13            34     ...
            1             8            21            55     ...
```

image: Spiral_of_Theodorus.png
Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4):
```

Selanjutnya dihubungan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
>
```

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memmerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
>function gstep (v) ...
```

```
  w=[-v[2];v[1]];
  return v+w/norm(w);
  endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

```
0.739085133216
```

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = \cos(x).$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

```
~0.739085133211,0.7390851332133~
```

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan x=cos(x).

```
>h=expand(hasil,100), cos(h) << h
```

```
~0.73908513309,0.73908513333~
1
```

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi x(n+1)=f(x(n)) akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

```
1.41421356237
1.41421356237
```

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

```
>iterate("f",2,5)
```

```
[2,  1.5,  1.41667,  1.41422,  1.41421,  1.41421]
```

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",~1,2~,5)
```

```
[ ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~ ]
```

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

$$x = \frac{1}{2}\left(x + \frac{2}{x}\right).$$

Satu-satunya solusi adalah

$$x = \sqrt{2}.$$

```
>iterate("s",~1,2~)
```

```
~1.41421356236,1.41421356239~
```

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1}, b_{n+1}) = \left(\frac{a_n + b_n}{2}, \sqrt{a_n b_n}\right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2;sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g",[1;5])
```

```
        2.60401
        2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g",[1;5],4)
```

```
              1             3       2.61803       2.60403       2.60401
              5       2.23607       2.59002       2.60399       2.60401
```

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g",[~1~;~5~],4)
```

```
  Interval 2 x 5 matrix

  ~0.999999999999999778,1.00000000000000022~     ...
  ~4.99999999999999911,5.00000000000000089~     ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)",2,10)
```

```
  [2,  1.41421,  1.18921,  1.09051,  1.04427,  1.0219,  1.01089,
  1.00543,  1.00271,  1.00135,  1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepatdengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)
```

```
[1.04888,  1.00028,  1,  1]
```

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

```
1.41421356237
```

Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

```
[1,  2,  6,  24,  120,  720,  5040,  40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
>plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
>x=[1;1]; repeat xnew=A.x-b; until all(xnew~=x); x=xnew; end; ...
>x,
```

```
-7.09677
-7.74194
```

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
```

```
x=x0;
maxiter=0;
repeat
  xnew=f$(x);
  maxiter=maxiter+1;
  until xnew~=x;
  x=xnew;
end;
return maxiter;
endfunction
```

Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

```
5
```

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

```
>x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ...
>  plot2d(x,hasil):
```

Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahnnya tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

```
>hasil[1:10]
```

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

```
>x[nonzeros(differences(hasil))]
```

```
[1.5,  2,  3.4,  6.6]
```

maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya 1.5, 2, 3.4, dan 6.6.

Contoh berikutnya adalah metode Newton pada polinomial kompleks berderajat 3.

```
>p &= x^3-1; newton &= x-p/diff(p,x); $newton
```

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslinya 10 kali).

```
>function iterasi(f$,x,n=10) ...
```

```
loop 1 to n; x=f$(x); end;
return x;
endfunction
```

Kita mulai dengan menentukan titik-titik grid pada bidang kompleksnya.

```
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
```

Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p)()
```

```
[ -0.5+0.866025i,  -0.5-0.866025i,  1+0i  ]
```

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

```
>C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ...
>  plot2d(none,-r,r,-r,r); plotrgb(C):
```

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

Error in:
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1) ...
                         ^
```

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor eskpresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi tersebut
```

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
>$sum(sin(k*x)/k,k,1,5)
```

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \ldots.$$

```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \le x \le 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmtable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n x^x di x=1.

```
>mxmtable("diffat(x^x,x=1,n)","n",1,8,frac=1);
```

```
      1
      2
      3
      8
     10
     54
    -42
    944
```

```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret secara simbol
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi
>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs(x)<1);
```

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

```
>$'sum(x^k/k!,k,0,inf)=ev(sum(x^k/k!,k,0,inf),simpsum=true)
>$limit(sum(x^k/k!,k,0,n),n,inf)
>function d(n) &= sum(1/(k^2-k),k,2,n); $'d(n)=d(n)
>$d(10)=ev(d(10),simpsum=true)
>$d(100)=ev(d(100),simpsum=true)
```

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

BAB 5

KB PEKAN 8: MENGGUNAKAN EMT UNTUK GEOMETRI

[a4paper,10pt]article eumat

Nama : Chintya Wijayanti
NIM : 22305144029
Kelas : Matematika E 2022

Aplikasi Komputer ** 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)ˆ2 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 poin dan plot mereka.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri meliputi fungsi untuk membuat garis dan lingkaran. Format garis adalah [a,b,c], yang mewakili garis dengan persamaan ax+by=c.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1,  2,  2]
```

Hitunglah garis tegak lurus yang melalui A pada BC.

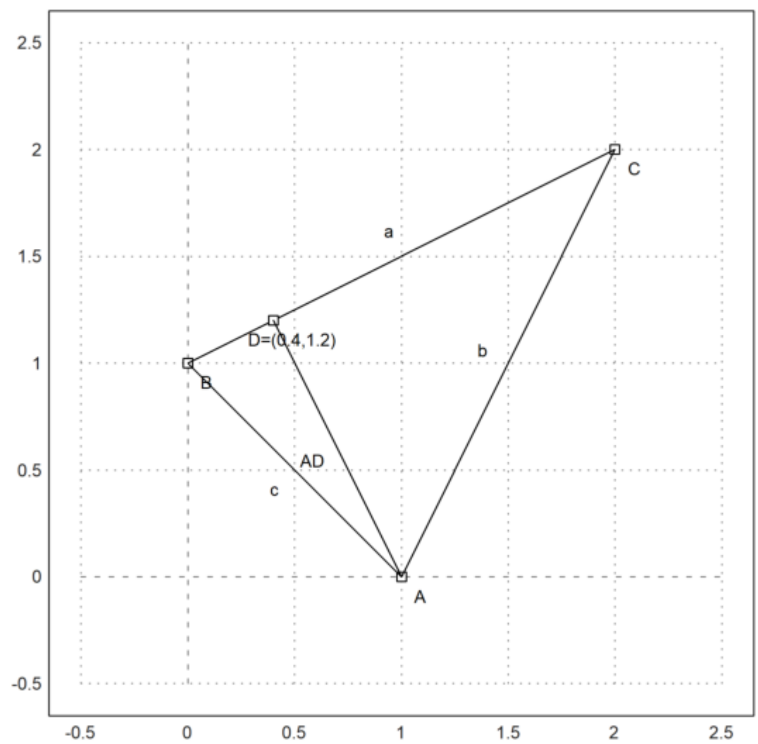
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD.BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

```
1.5
```

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langusng dengan fungsi
```

```
1.5
```

Cara lain menghitung luas segitigas ABC:

```
>distance(A,D)*distance(B,C)/2
```

```
1.5
```

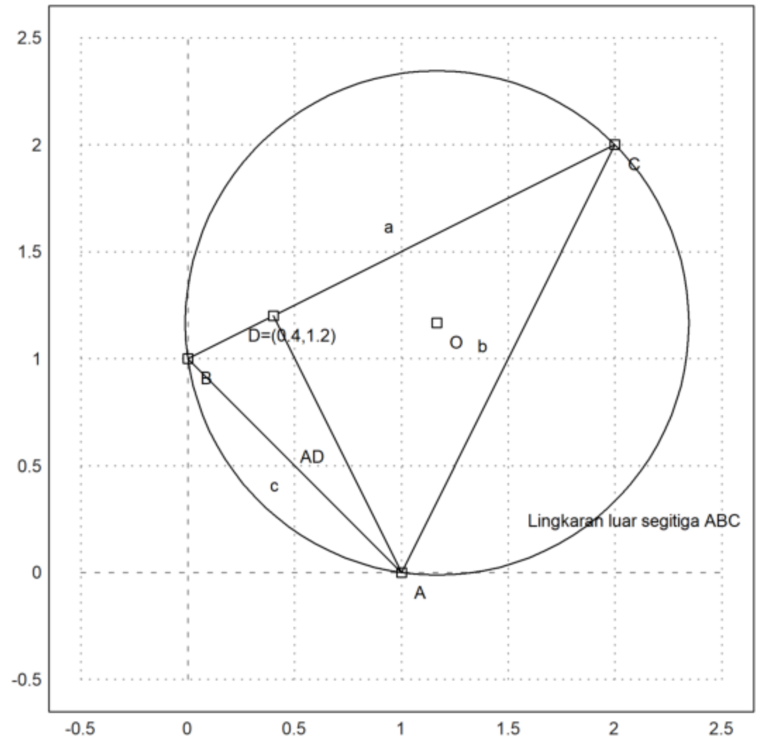
Sudut di C

```
>degprint(computeAngle(B,C,A))
```

```
36°52’11.63’’
```

Sekarang lingkaran luar segitiga.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```



Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
[1.16667,  1.16667]
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

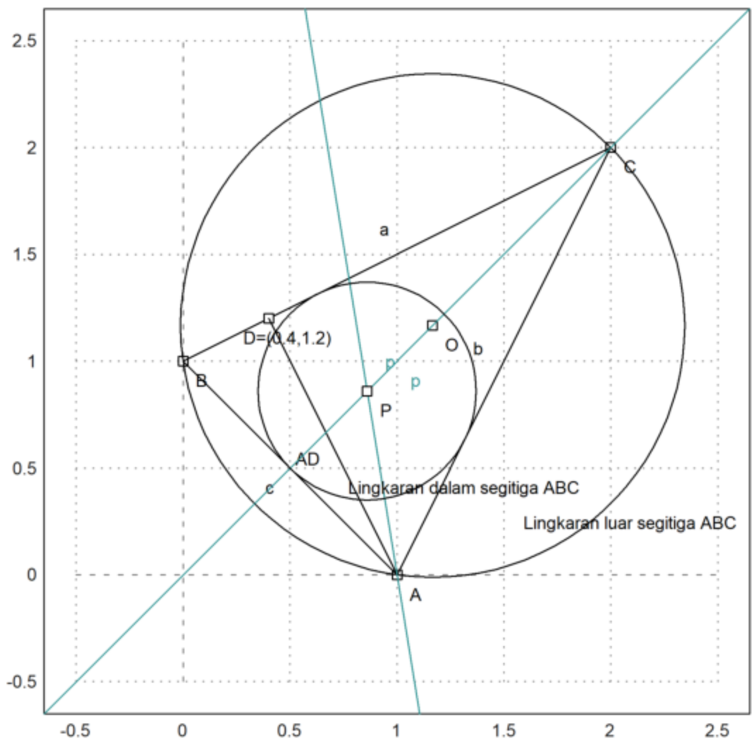
```
[0.86038,  0.86038]
```

Tambahkan semuanya ke plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gambar lingkaran dalam
```



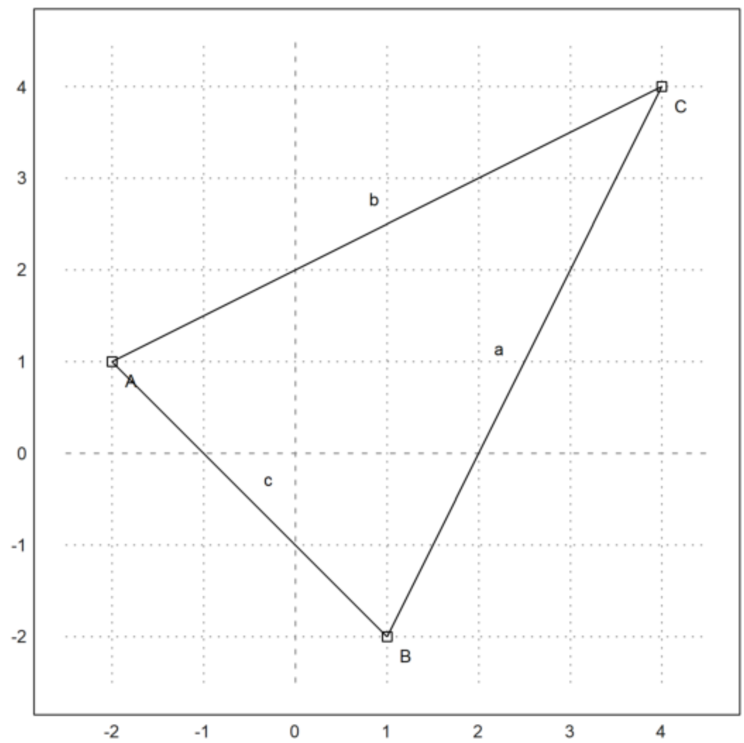
Latihan

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.

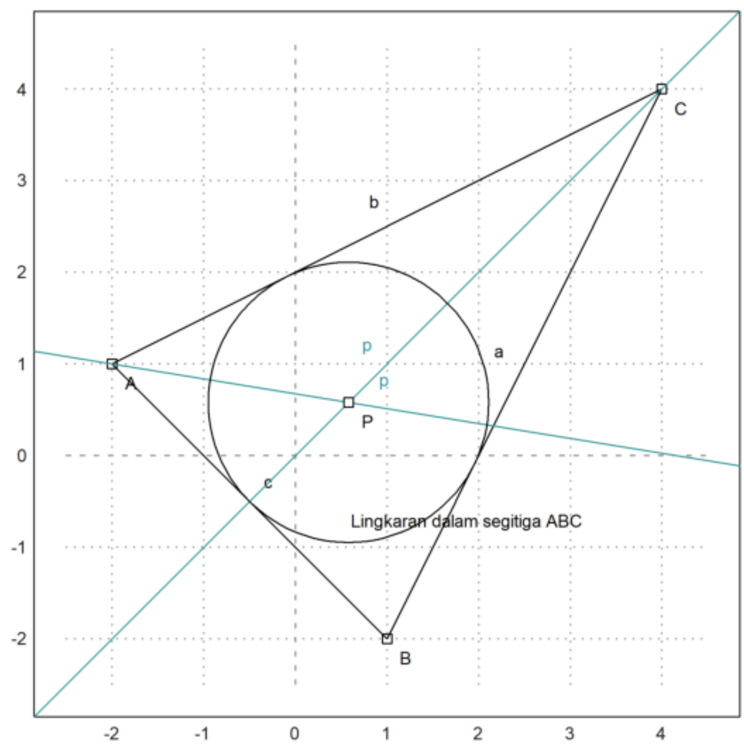
```
>l=angleBisector(A,C,B);
>g=angleBisector(C,A,B);
>P=lineIntersection(l,g)
```

```
[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"):
```



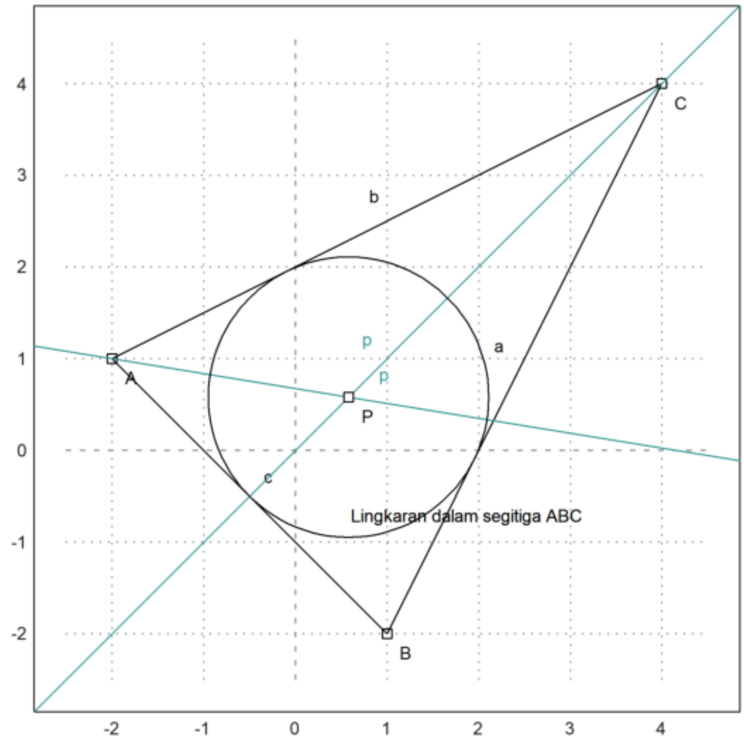
Jadi, terbukti bahwa garis bagi sudut yang ketiga juga melalui titik pusat lingkaran dalam.

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 Simbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan perhitungan simbolis sekarang.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi memberikan perhitungan simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

```
                        [- 1, 2, 2]
```

Kita bisa mendapatkan persamaan garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$2\,y-x=2$$

$$\left[y=\frac{x}{2}+1\right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$\left(x_1-1\right)\,y-x\,y_1=-y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

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

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

```
                             2  6
                            [-, -]
                             5  5
```

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[\frac{2}{5},\frac{6}{5}\right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{6},\frac{7}{6},\frac{5}{3\,\sqrt{2}}\right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{3\sqrt{2}}$$

$$1.178511301977579$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi sudu
```

$$\left[\frac{\sqrt{2}\,\sqrt{5}+2}{6}, \frac{\sqrt{2}\,\sqrt{5}+2}{6}\right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
[0.86038,  0.86038]
```

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

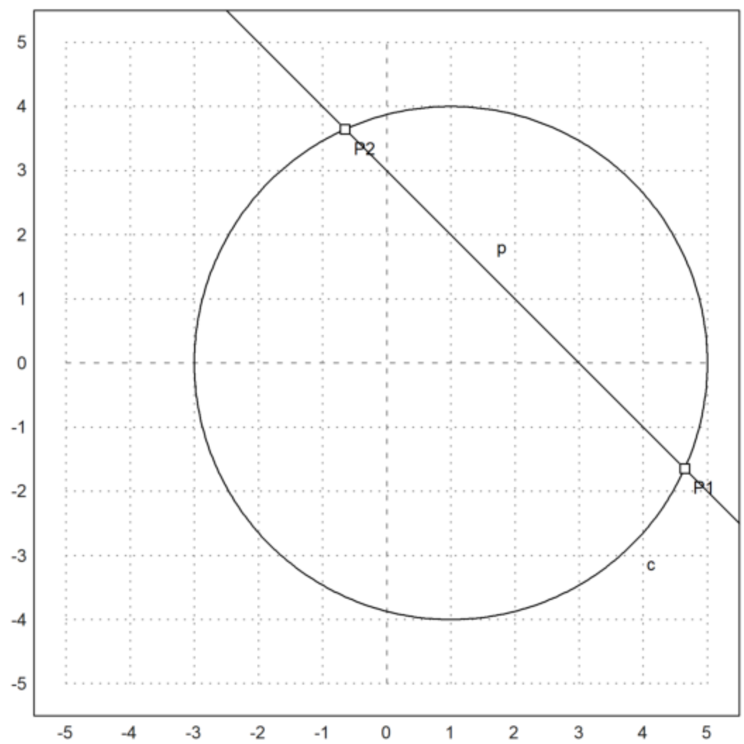
```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik potong.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2,
```

```
[4.64575,  -1.64575]
[-0.645751,  3.64575]
```

```
>plotPoint(P1); plotPoint(P2):
```



Begitu pula di Maxima.

```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

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

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[\left[\sqrt{7}+2, 1-\sqrt{7}\right], \left[2-\sqrt{7}, \sqrt{7}+1\right]\right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

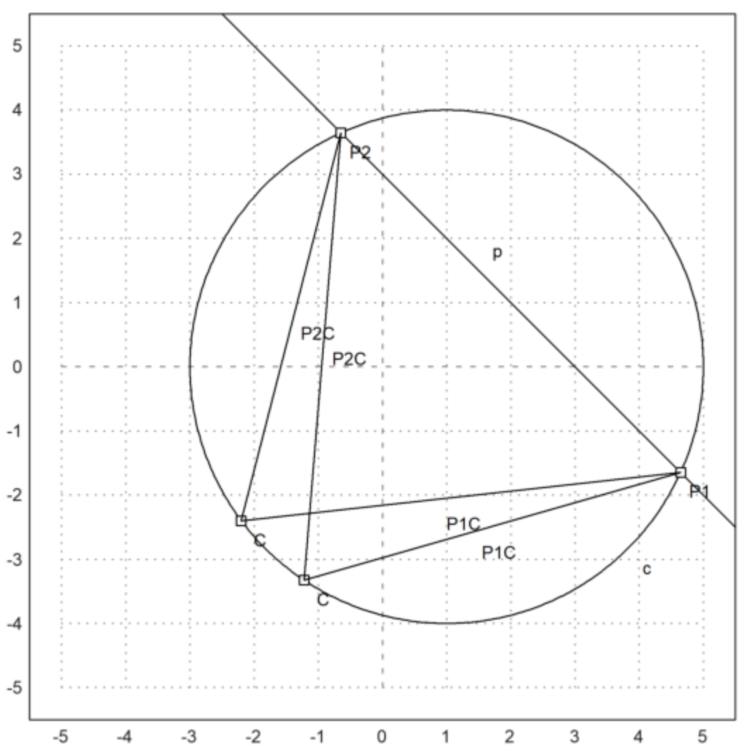
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

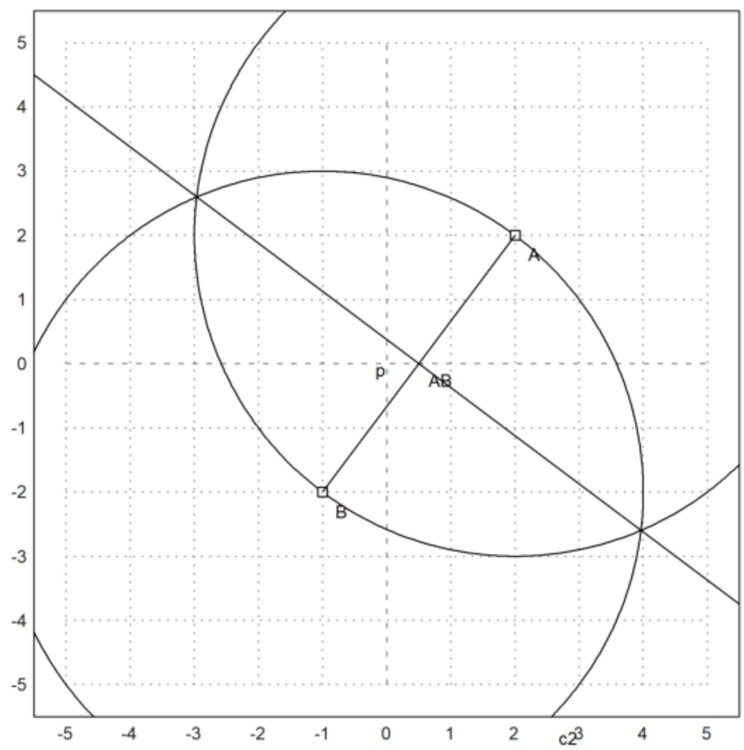
```
>insimg;
```



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, kami melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk persimpangan cukup terlibat. Tetapi kita dapat menyederhanakannya, jika kita memecahkan y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y=\frac{\left(b_2-a_2\right)\,x-a_1\,b_2+a_2\,b_1}{b_1-a_1}\right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1-a_1)}{(b_2-a_2)}\times\frac{(b_2-a_2)}{(b_1-a_1)}=-1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L=\sqrt{s(s-a)(s-b)(s-c)}\quad \text{dengan } s=(a+b+c)/2,$$

Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC}=\frac{1}{2}a\times y.$$

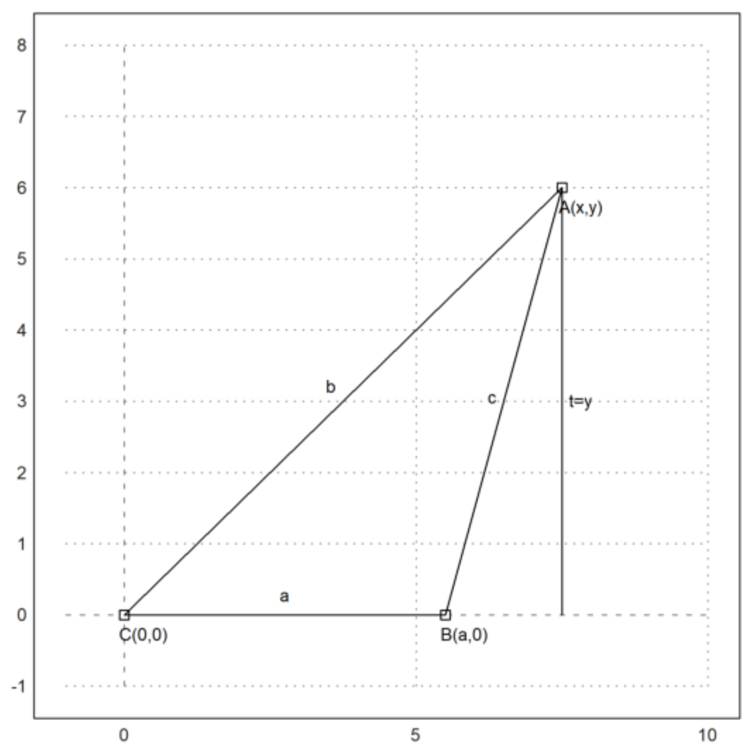
Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

[]

```
>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):
```




```
>sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

[]

Ekstrak solusi y.

```
>ysol &= y with sol[2][2]; $ysol
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
ysol &= y with sol[2][2]; $ysol ...
                        ^
```

Kami mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H\left(a, b, [1, 0, 4]\right) = \frac{|a|\;|ysol|}{2}$$

```
>$'Luas=H(3,4,5) // luas segitiga dengan panjang sisi-sisi 3, 4, 5
```

$$Luas = \frac{3\;|ysol|}{2}$$

Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
Variable or function ysol not found.
Try "trace errors" to inspect local variables after errors.
H:
    useglobal; return abs(a)*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
        ^
```

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan dua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga sengan panjang sisi 3, 4, x (1<= x <=7)
```

```
Variable or function ysol not found.
Error in expression: 3*abs(ysol)/2
%ploteval:
    y0=f$(x[1],args());
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

Kasus umum juga berfungsi.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
 -- an error. To debug this try: debugmode(true);

Error in:
$solve(diff(H(a,b,c)^2,c)=0,c) ...
                              ^
```

Sekarang mari kita cari himpunan semua titik di mana b+c=d untuk beberapa konstanta d. Diketahui bahwa ini adalah elips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...
                         ^
```

Dan buat fungsi ini.

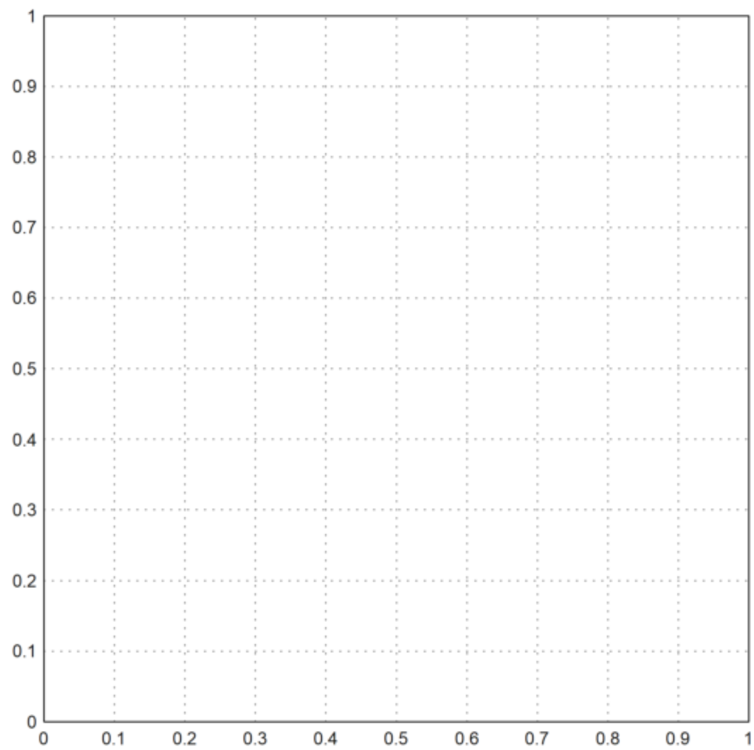
```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

$$0$$

$$0$$

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan 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 setengah sumbu.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{a^2}{d^2}$$

Kita lihat bahwa tinggi dan luas segitiga adalah maksimal untuk x=0. Jadi luas segitiga dengan a+b+c=d maksimal jika segitiga sama sisi. Kami ingin menurunkan ini secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[\frac{a\,ysol^2}{2}=0,0=0\right]$$

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusinya a=b=c=d/3.

```
>$solve(eqns,[a,b])
```

$$[[a=0,b=\%r_1]]$$

Ada juga metode Lagrange, memaksimalkan H(a,b,c)ˆ2 terhadap a+b+d=d.

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
>   diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
 -- an error. To debug this try: debugmode(true);

Error in:
... la,    diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...
                                                     ^
```

Kita bisa membuat plot situasinya

Pertama-tama atur poin di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
A &= at([x,y],sol[2]); $A ...
                     ^
```

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

$$[0,0]$$

$$[a,0]$$

Kemudian atur rentang plot, dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
>plotPoint(mxmeval("A"),"A"):
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
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.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("A")):
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval("B ...
                        ^
```

Hitung tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat lingkaran.

```
>U &= lineIntersection(h,g);
```

Kami mendapatkan rumus untuk jari-jari lingkaran.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{a_2^2 + a_1^2}\,\sqrt{a_2^2 + a_1^2 - 2\,a\,a_1 + a^2}}{2\,|a_2|}$$

Mari kita tambahkan ini ke plot.

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2,(a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
             ^
```

Menggunakan geometri, kami memperoleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radiusnya. Kami dapat memeriksa, apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita kuadratkan.

```
>$c^2/sin(computeAngle(A,B,C))^2  | factor
```

$$\left[\frac{a_2^2+a_1^2}{a_2^2},0,\frac{16\,\left(a_2^2+a_1^2\right)}{a_2^2}\right]$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

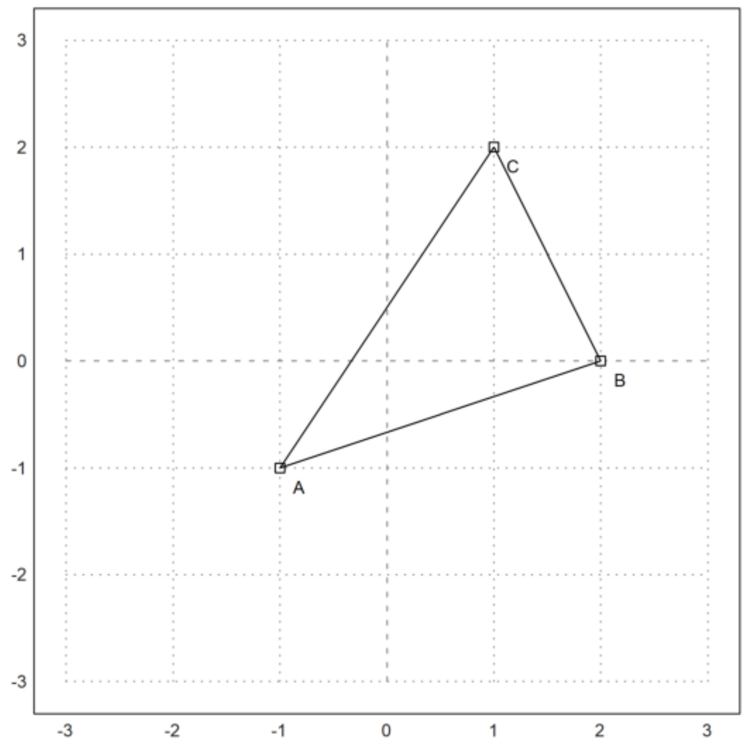
```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan titik ke sana. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kita juga bisa menambahkan sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
```



Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

$$-\frac{7}{2}$$

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

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

Dan juga dapatkan rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

$$3\,y-x=-2$$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{3\,y-x+2}{\sqrt{10}}$$
$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

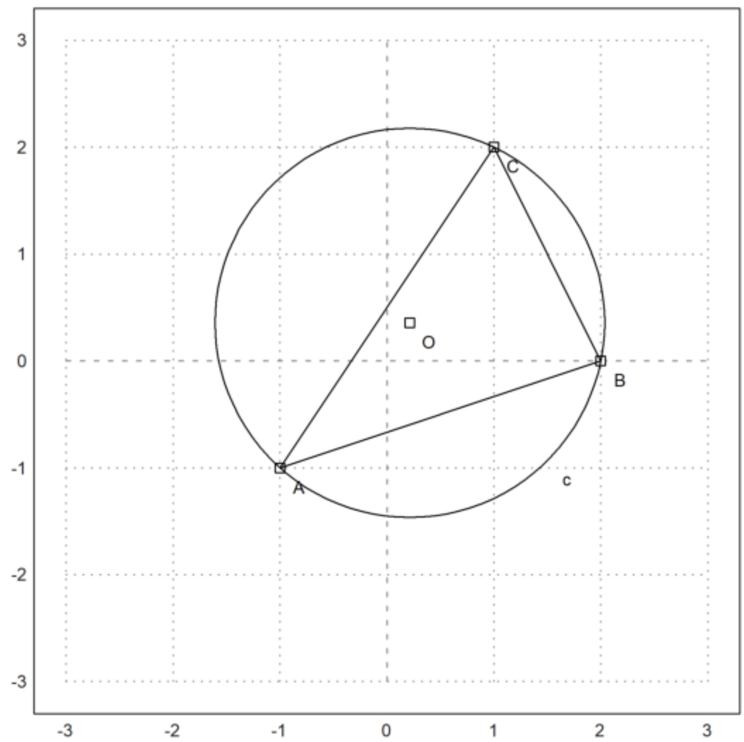
$$\left(y-\frac{5}{14}\right)^2+\left(x-\frac{3}{14}\right)^2=\frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14},\frac{5}{14}\right]$$

Gambarkan lingkaran dan pusatnya. Cu dan U adalah simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```



Kita dapat menghitung perpotongan ketinggian di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
>  perpendicular(B,lineThrough(A,C))); $H
```

$$\left[\frac{11}{7}, \frac{2}{7}\right]$$

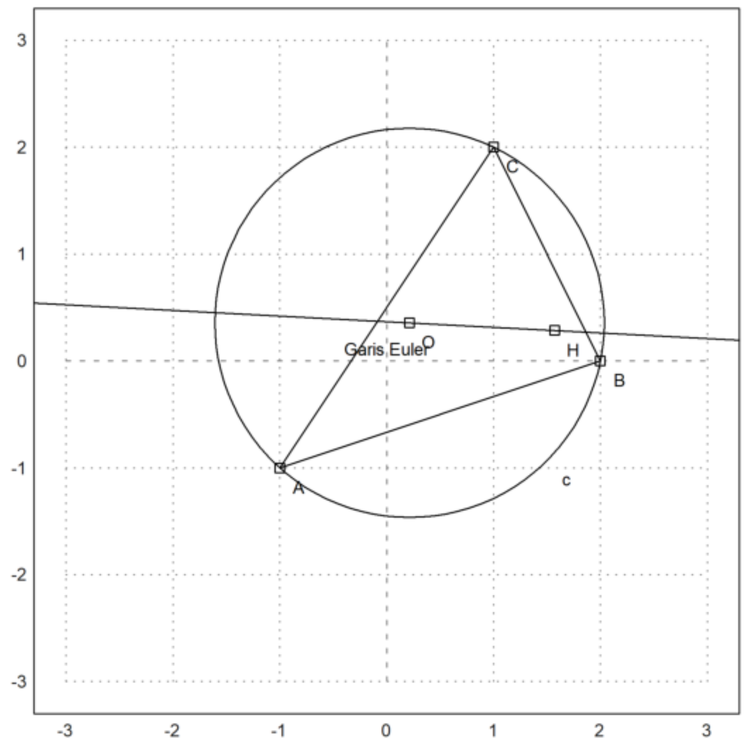
Sekarang kita dapat menghitung garis Euler dari segitiga.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19\,y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kami.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

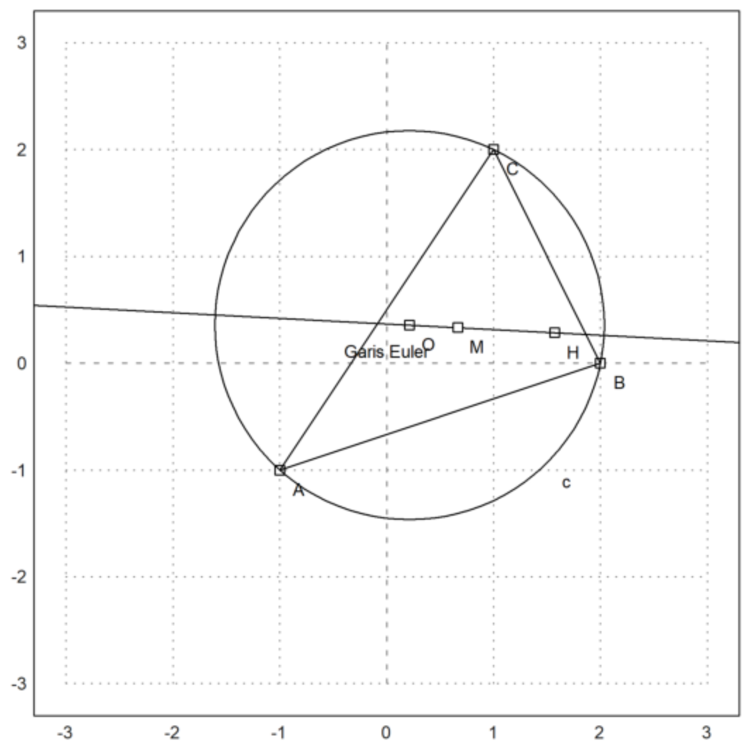


Pusat gravitasi harus berada di garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M"): // titik berat
```



Teorinya memberitahu kita MH=2*MO. Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
>$distance(M,H)/distance(M,O)|radcan
```

$$2$$

Fungsi termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\,\sqrt{13}}\right)$$

```
60°15'18.43''
```

Persamaan untuk pusat incircle tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

$$\left[\frac{\left(2^{\frac{3}{2}}+1\right)\,\sqrt{5}\,\sqrt{13}-15\,\sqrt{2}+3}{14} , \frac{\left(\sqrt{2}-3\right)\,\sqrt{5}\,\sqrt{13}+5\,2^{\frac{3}{2}}+5}{14}\right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

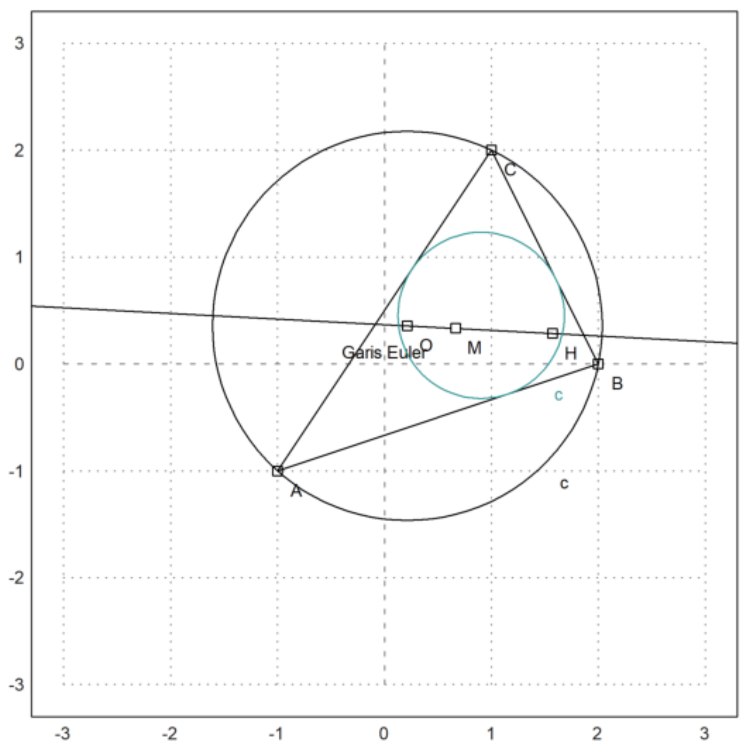
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{\left(-41\,\sqrt{2}-31\right)\,\sqrt{5}\,\sqrt{13}+115\,\sqrt{2}+614}}{7\,\sqrt{2}}$$

```
>LD &=  circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
>color(5); plotCircle(LD()):
```



Parabola

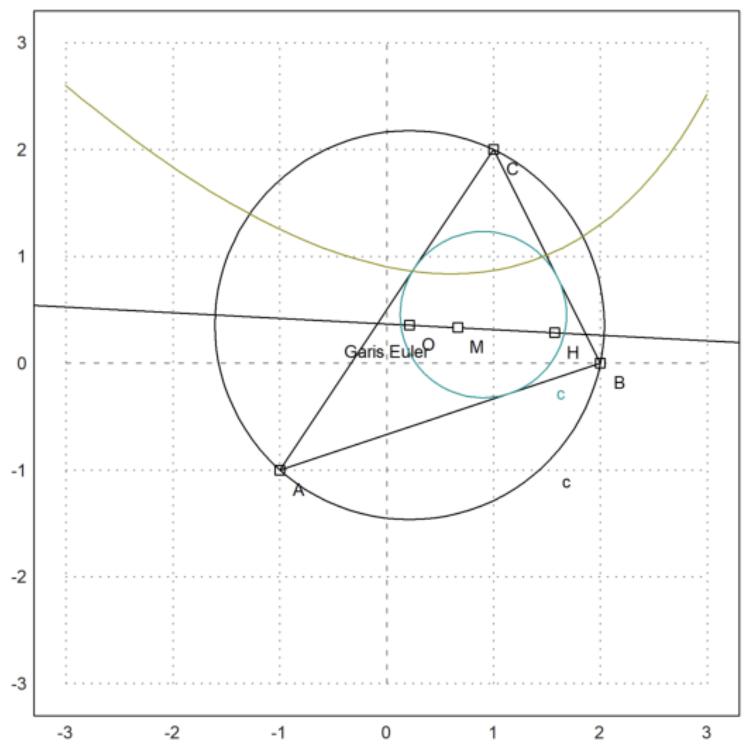
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3\,y-x+2}{\sqrt{10}}-\sqrt{\left(2-y\right)^2+\left(1-x\right)^2}=0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya menjadi beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

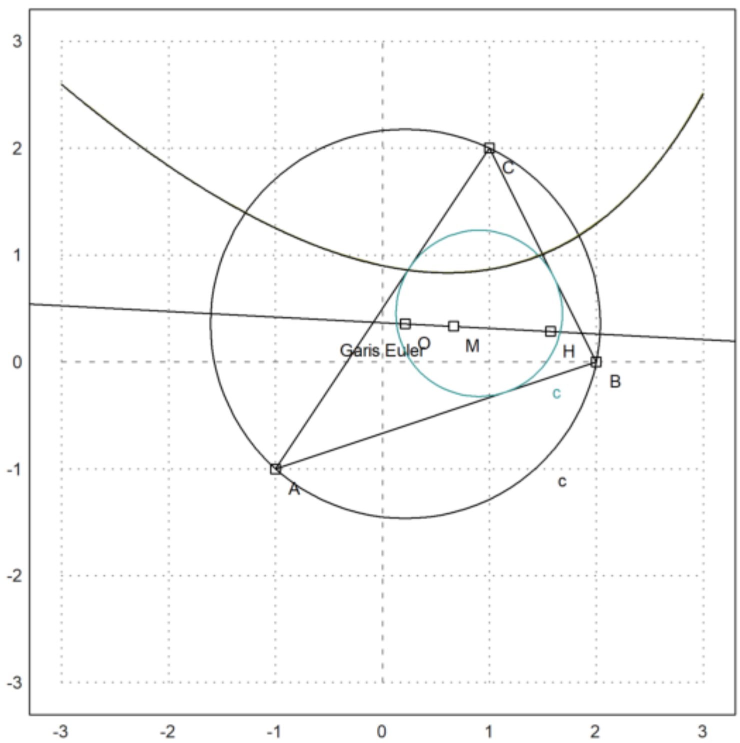
```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
                        y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

maxima: akar[1]

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```




```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g\left(x\right)=-3\,x-\sqrt{70}\,\sqrt{9-2\,x}+26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$\sqrt{1503-54\,\sqrt{11}\,\sqrt{70}}$$

$$2.135605779339061$$

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[\frac{80-3\sqrt{11}\sqrt{70}}{10},\frac{20-\sqrt{11}\sqrt{70}}{10}\right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jatak T ke AB
```

$$\sqrt{1503-54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

Contoh 5: Trigonometri Rasional

Ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti pengertian klasik tentang jarak dan sudut dengan kuadrat dan penyebaran. Dengan menggunakan ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

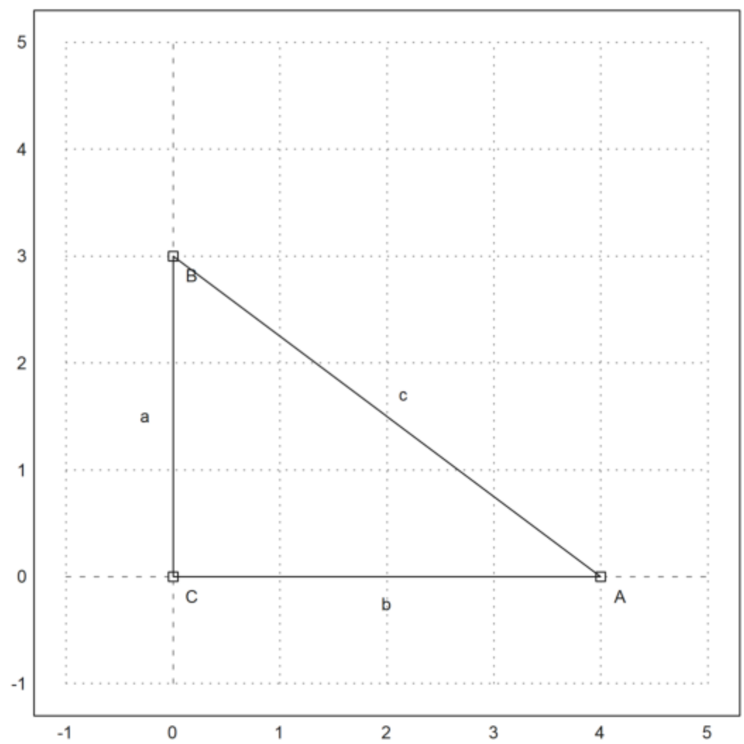
Berikut ini, saya memperkenalkan konsep, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasil tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolis sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk merencanakan geometri bidang yang terdapat dalam file Euler ”geometry.e”.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana wa adalah sudut di A. Cara yang biasa untuk menghitung sudut ini, adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

```
>wa := arcsin(3/5); degprint(wa)
```

```
36°52'11.63''
```

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanya jarak kuadrat. Berikut ini, a, b, dan c menunjukkan kuadrat dari sisi-sisinya.

Teorema Pythogoras menjadi a+b=c.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

```
25 = 25
```

Pengertian kedua dari trigonometri rasional adalah penyebaran. Spread mengukur pembukaan antar baris. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai:

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadrat dari sembarang segitiga siku-siku dengan salah satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

```
9/25
```

Hukum kosinus trgonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c + b - a)^2 = 4bc\,(1 - s_a)$$

Di sini a, b, dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran sudut A. Sisi a, seperti biasa, berhadapan dengan sudut A.

Hukum ini diimplementasikan dalam file geometri.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[\left(bb-aa+\frac{7}{6}\right)^2,\left(bb-aa+\frac{7}{6}\right)^2,\left(bb-aa+\frac{5}{3\sqrt{2}}\right)^2\right]=\left[\frac{14\,bb\,(1-saa)}{3},\frac{14\,bb\,(1-saa)}{3},\frac{5\,2^{\frac{3}{2}}\,bb\,(1-saa)}{3}\right]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024=1024$$

Mari kita gunakan crosslaw ini untuk mencari spread di A. Untuk melakukan ini, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk spread yang tidak diketahui sa.

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memilikinya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$1024=1600\,(1-x)$$

$$\left[x=\frac{9}{25}\right]$$

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum a,b,c. Hasilnya adalah rumus yang menghitung penyebaran sudut segitiga yang diberikan kuadrat dari ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[\left[\frac{168\,bb\,x+36\,bb^2+(-72\,aa-84)\,bb+36\,aa^2-84\,aa+49}{36},\frac{168\,bb\,x+36\,bb^2+(-72\,aa-84)\,bb+36\,aa^2-84\,aa+49}{36},\right.\right.$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e dari Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak begitu mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

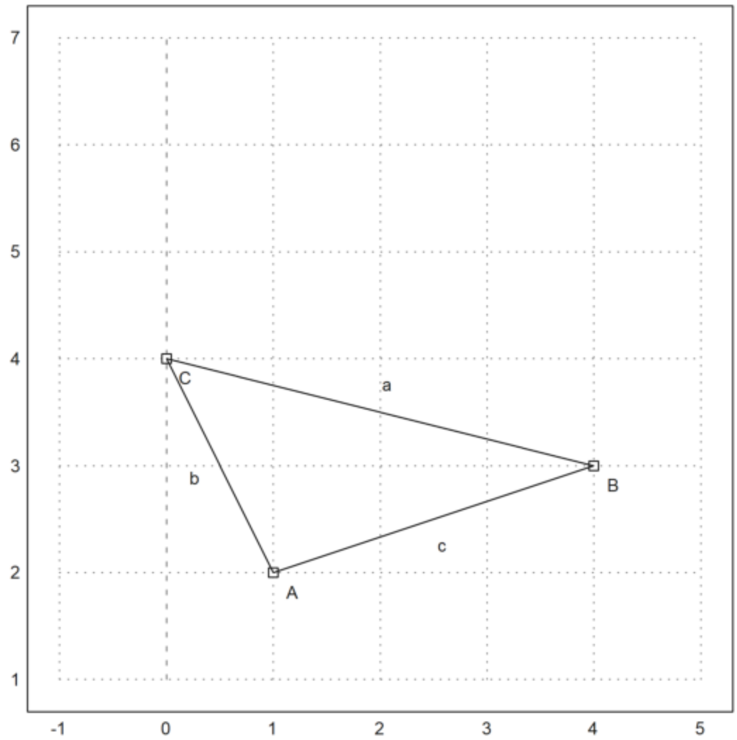
```
67°47'32.44''
```

Contoh lain

Sekarang, mari kita coba contoh yang lebih maju.

Kami mengatur tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
>setPlotRange(-1,5,1,7); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Menggunakan Pythogoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan jarak fungsi file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena c+b bukan a, maka segitiga itu bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$10$$

$$5$$

$$17$$

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali titik dua vektor. Hasilnya adalah beberapa pendekatan floating point.

$$A =< 1, 2 > \quad B =< 4, 3 >, \quad C =< 0, 4 >$$

$$\mathbf{a} = C - B =< -4, 1 >, \quad \mathbf{c} = A - B =< -3, -1 >, \quad \beta = \angle ABC$$

$$\mathbf{a.c} = |\mathbf{a}|.|\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a.c}}{|\mathbf{a}|.|\mathbf{c}|} = \frac{12 - 1}{\sqrt{17}\sqrt{10}} = \frac{11}{\sqrt{17}\sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\,\sqrt{17}}\right)$$

```
32.4711922908
```

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan x.

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

$$4 = 200\,(1 - x)$$

$$\left[x = \frac{49}{50}\right]$$

Yaitu, apa yang dilakukan oleh penyebaran fungsi yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Itu menyelesaikan istilah sin(arccos(...)) menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

Menurut definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadrat dan menyebar.

image: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadratnya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berhadapan dengan kuadrat!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

Rumus Bangau

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama kita hitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat (”quadrea”?), faktorkan dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{-c^4-\left(-2\,b^2-2\,a^2\right)\,c^2-b^4+2\,a^2\,b^2-a^4}{4\,a^2\,c^2}$$

$$\frac{\left(-c+b+a\right)\,\left(c-b+a\right)\,\left(c+b-a\right)\,\left(c+b+a\right)}{16}$$

Aturan Triple Spread

Kerugian dari spread adalah mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread dari sebuah segitiga memenuhi aturan ”triple spread” berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$\left(sc+sb+sa\right)^2=2\,\left(sc^2+sb^2+sa^2\right)+4\,sa\,sb\,sc$$

Aturan ini berlaku untuk setiap tiga sudut yang menambah 180 °.

$$\alpha+\beta+\gamma=\pi$$

Sejak menyebar

$$\alpha,\pi-\alpha$$

sama, aturan triple spread juga benar, jika

$$\alpha+\beta=\gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran rangkap tiga juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung penyebaran sudut 60°. Ini 3/4. Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[x = \frac{3}{4}, x = 0\right]$$

Sebaran 90° jelas 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita mendapatkan a+b=1.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$(y + x + 1)^2 = 2\,\left(y^2 + x^2 + 1\right) + 4\,x\,y$$

$$[x = 1 - y]$$

Karena sebaran 180°-t sama dengan sebaran t, rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami membuat ini fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

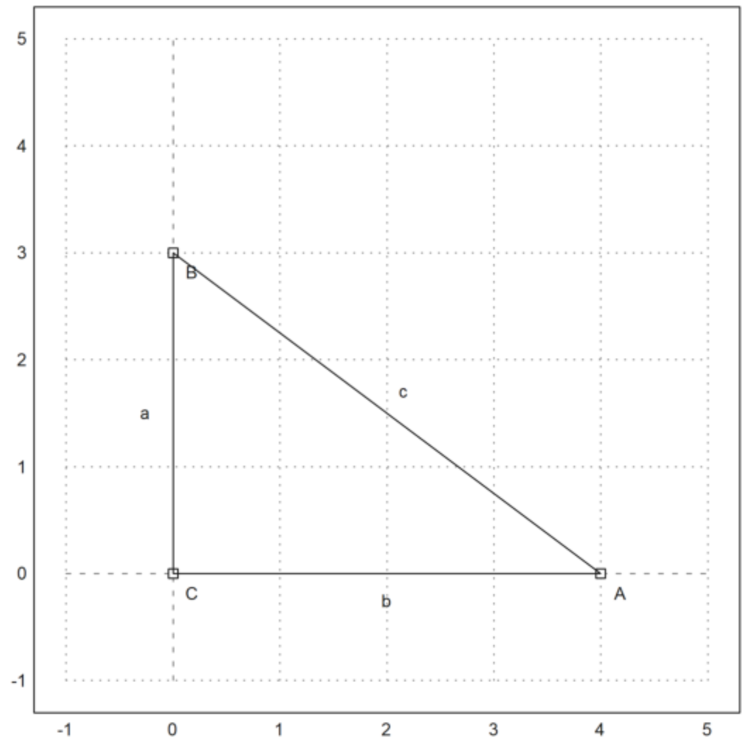
$$\left[x = 4\,a - 4\,a^2, x = 0\right]$$

```
- 4 (a - 1) a
```

Pembagi Sudut

Ini situasinya, kita sudah tahu.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaikannya untuk umum a,b,c.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung penyebaran sudut yang dibagi dua di A, dengan menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut terbelah 180 °-wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

$$\left(2\,x+\frac{a}{b+a}\right)^2=2\,\left(2\,x^2+\frac{a^2}{\left(b+a\right)^2}\right)+\frac{4\,a\,x^2}{b+a}$$

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

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kami dapat mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

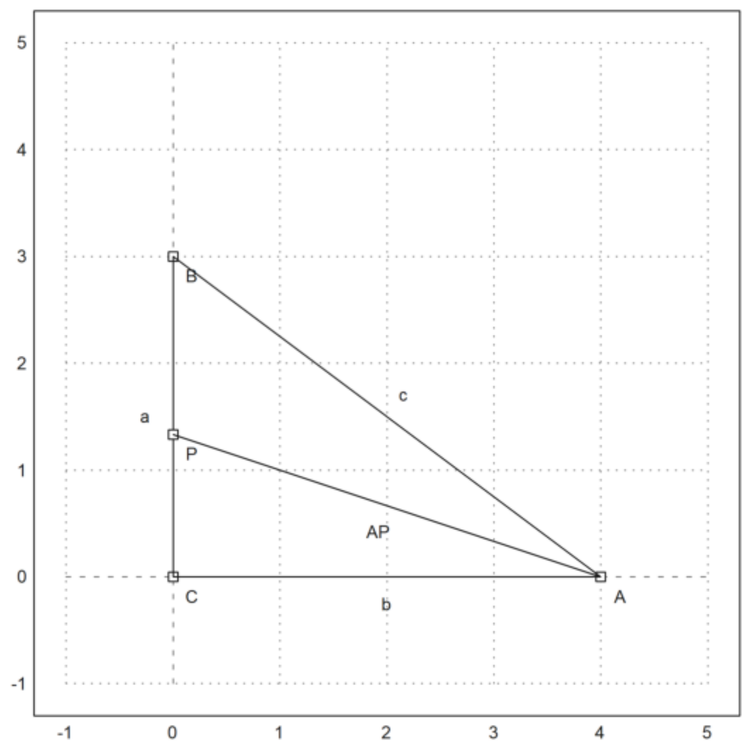
```
18°26'5.82''
```

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
[0,  1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```



Mari kita periksa sudut dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku dalam segitiga apa pun. Kuadratkan, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a,b,c menunjukkan qudrances.

Karena spread CPA adalah 1-sa2, kita dapatkan darinya bisa/1=b/(1-sa2) dan dapat menghitung bisa (kuadran dari garis-bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2\,b\,\left(b+a\right)}{\sqrt{b}\,\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b\,\left(\sqrt{b}\,\sqrt{b+a}-b-a\right)}{\sqrt{b}\,\sqrt{b+a}+b+a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

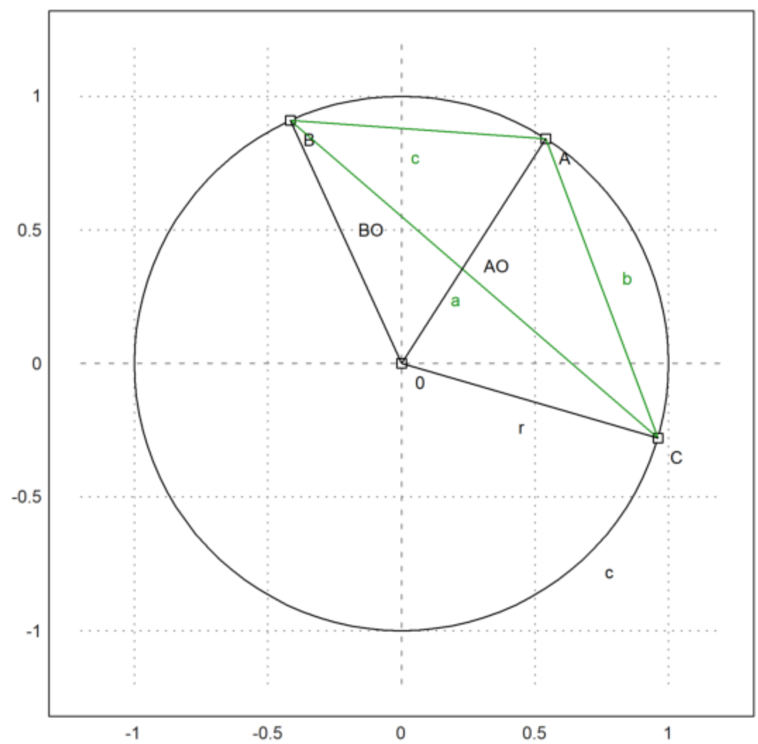
```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
1.33333333333
```

Sudut Akord

Perhatikan situasi berikut.

```
>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>color(1); O:=[0,0];  plotPoint(O,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```



Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadrat dari sisi.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

$$-\frac{a\,b\,c}{c^2-2\,b\,c+a\,\left(-2\,c-2\,b\right)+b^2+a^2}$$

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk poin A,B,C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jarinya memang 1.

```
>periradius(a,b,c)
```

```
1
```

Faktanya, spread CBA hanya bergantung pada b dan c. Ini adalah teorema sudut chord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Sebenarnya spreadnya adalah b/(4r), dan kita melihat bahwa sudut chord dari chord b adalah setengah dari sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

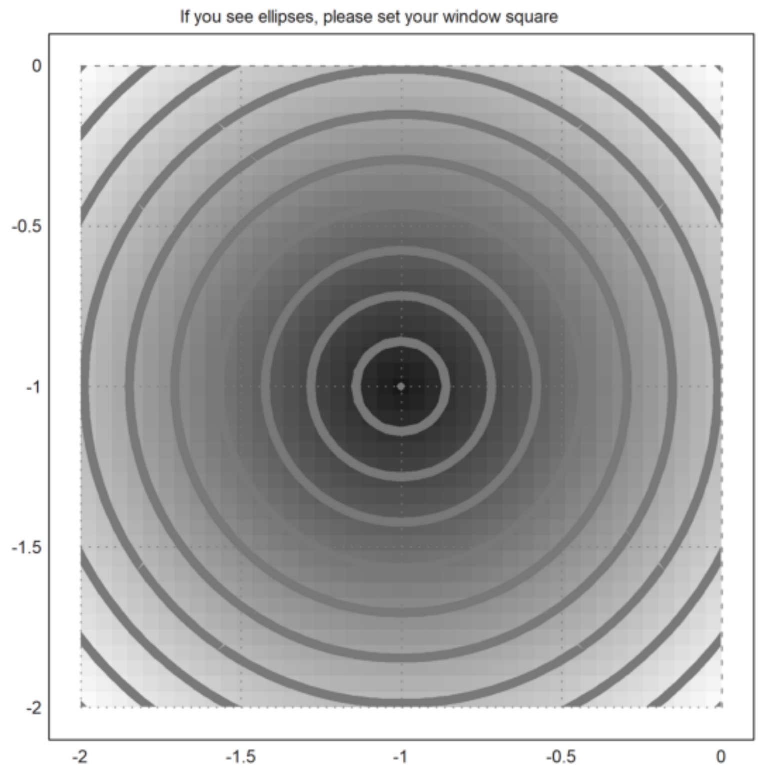
$$0$$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

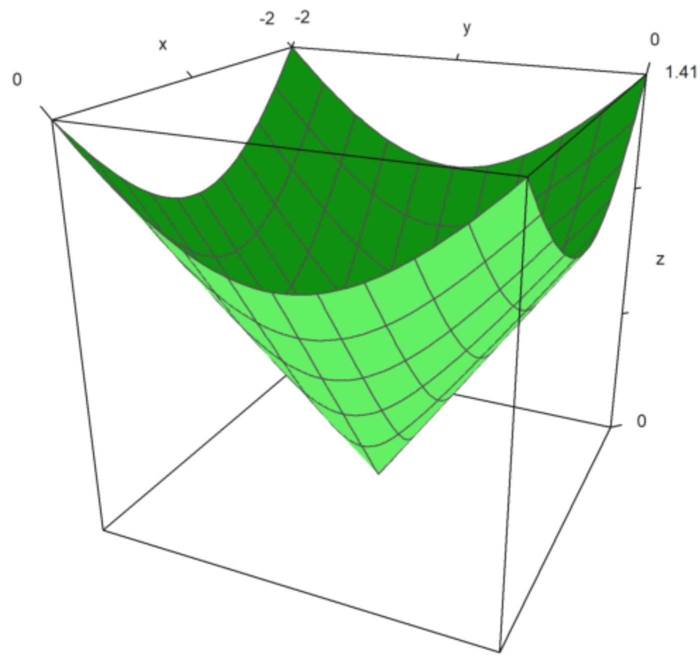
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

```
>&remvalue();
>A=[-1,-1];
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
>title="If you see ellipses, please set your window square"):
```



dan grafiknya juga agak sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

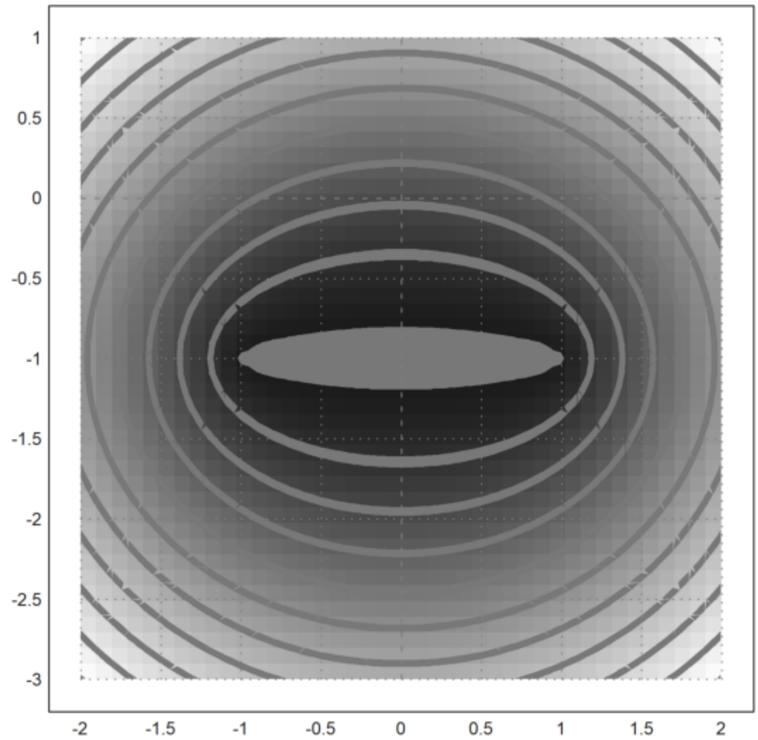


Tentu saja minimal 0 dicapai di A.

Dua poin

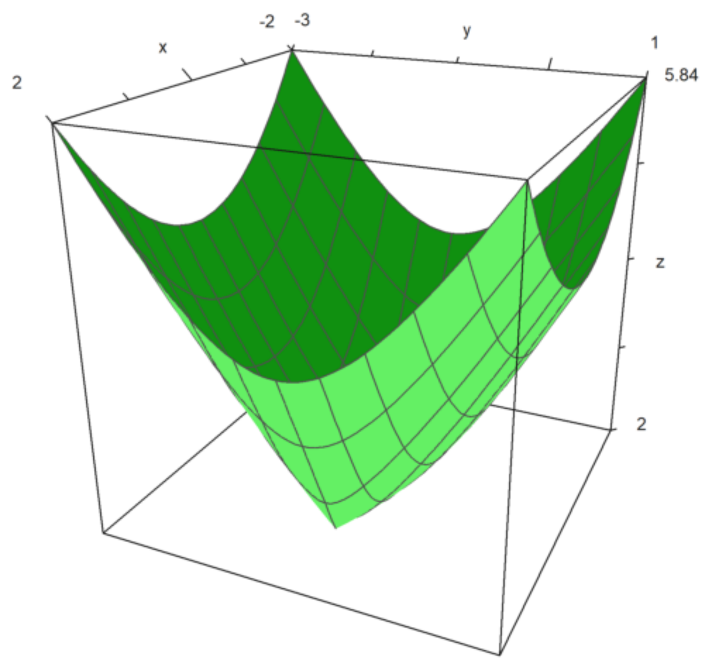
Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap). Ini adalah ”fakta yang diketahui” bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk AB minimum yang konstan pada segmen [AB]:

```
>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



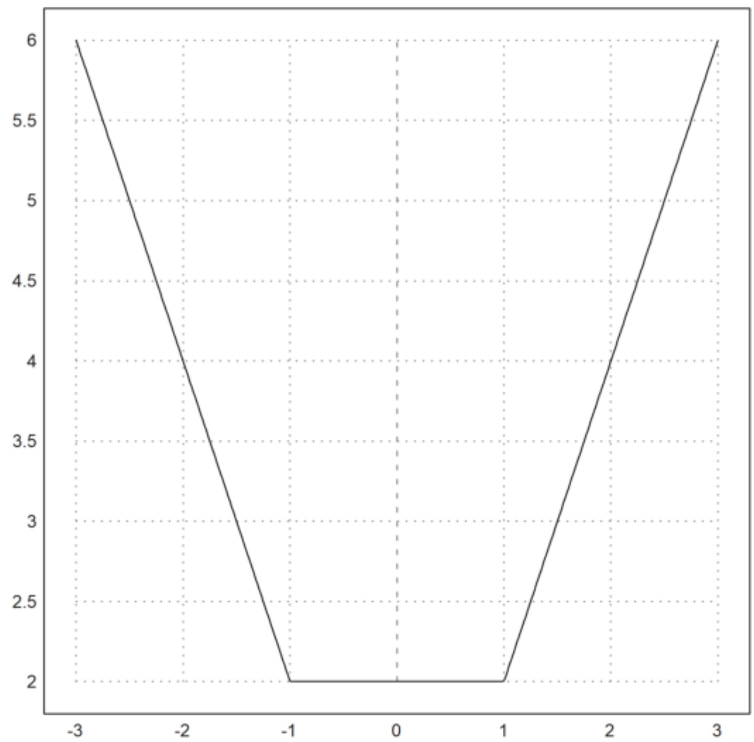
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



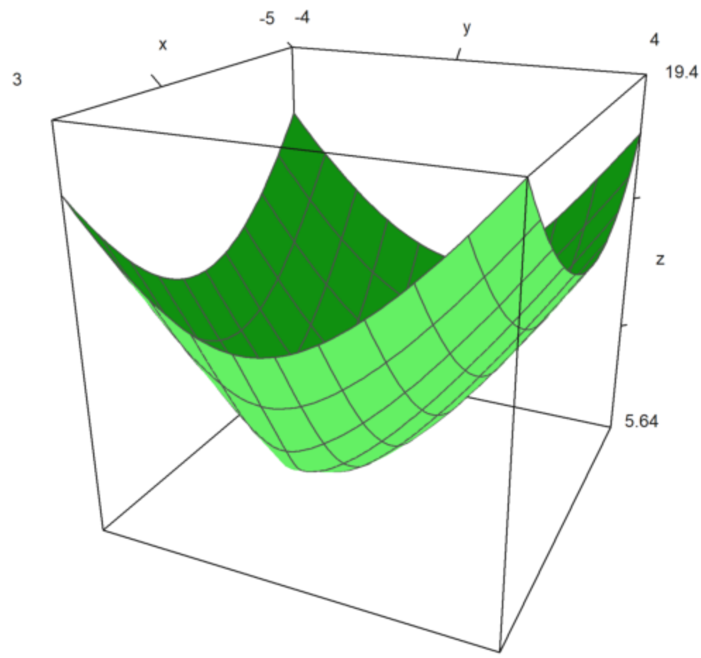
Tiga poin

Sekarang hal-hal yang kurang sederhana: Ini sedikit kurang terkenal bahwa MA+MB+MC mencapai minimum pada satu titik pesawat tetapi untuk menentukan itu kurang sederhana:

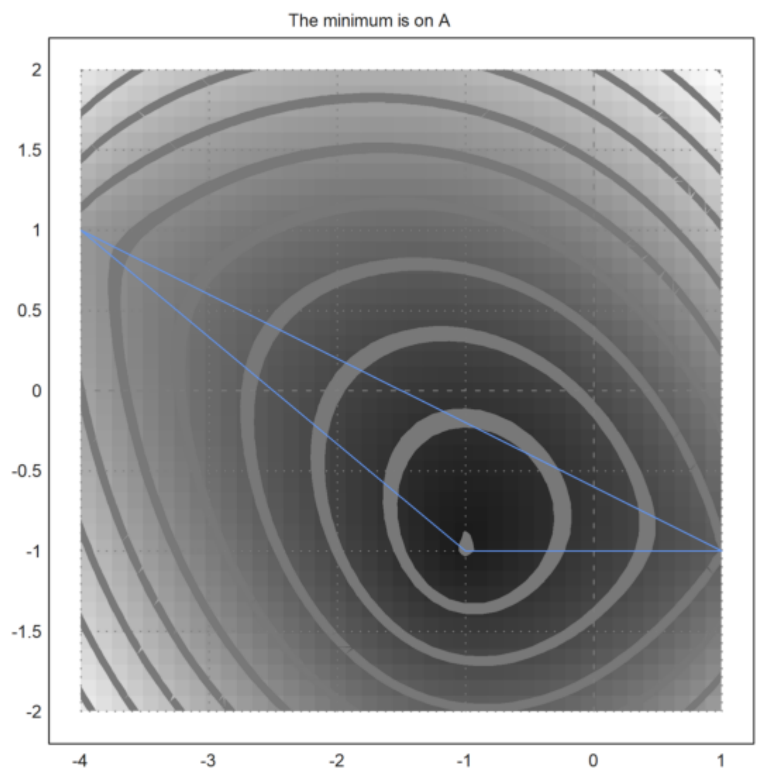
1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka minimum dicapai pada titik ini (misalnya AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```

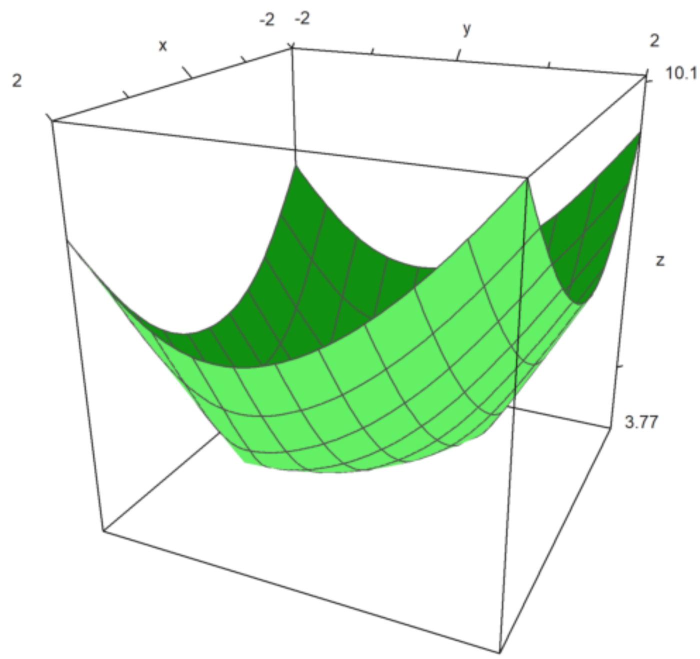



```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

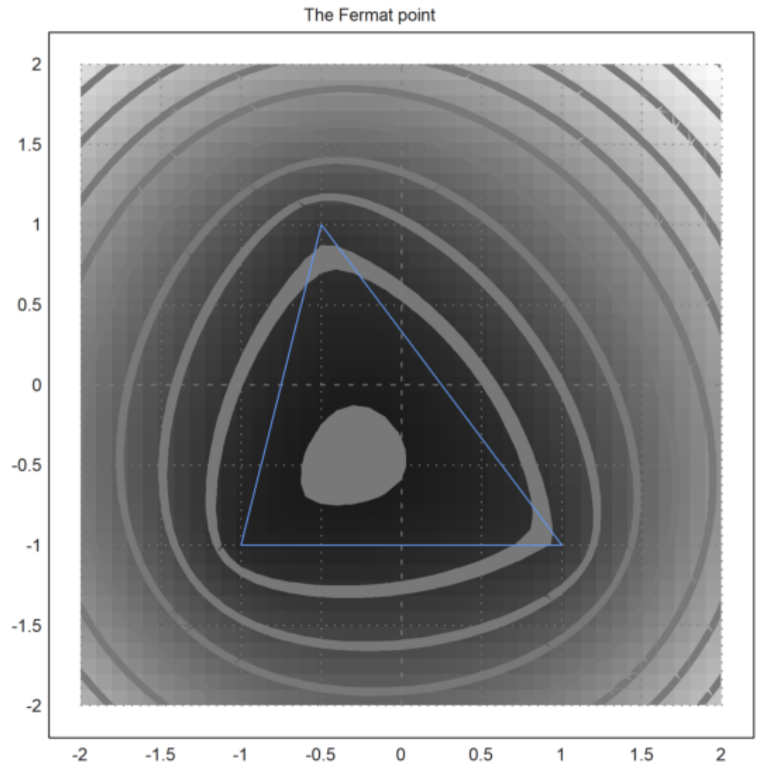


2) Tetapi jika semua sudut segitiga ABC kurang dari 120 °, minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing 120 °):

```
>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;
```



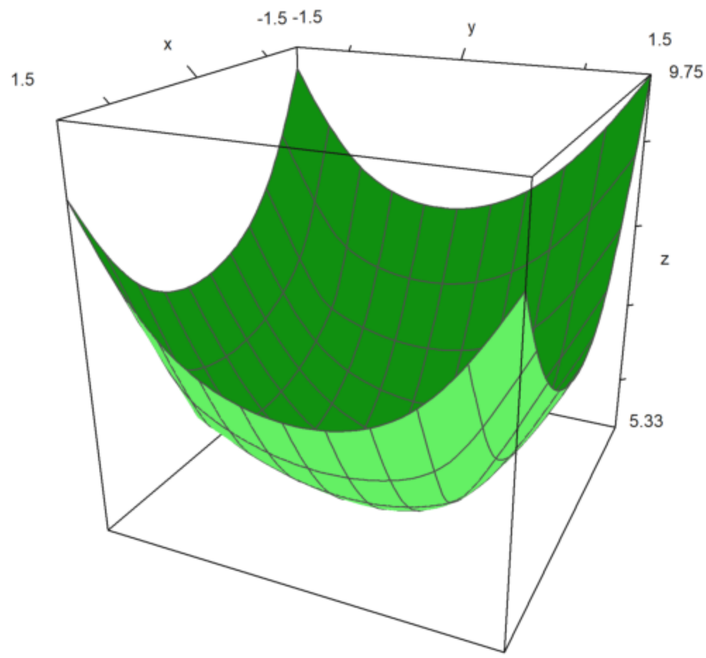
Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft yang ditulis di Jawa yang memiliki instruksi "garis kontur" ...

Semua ini di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada dilettants lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga FA+FB+FC minimal, disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torriccelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun tradisinya adalah mencatat poin ini F...

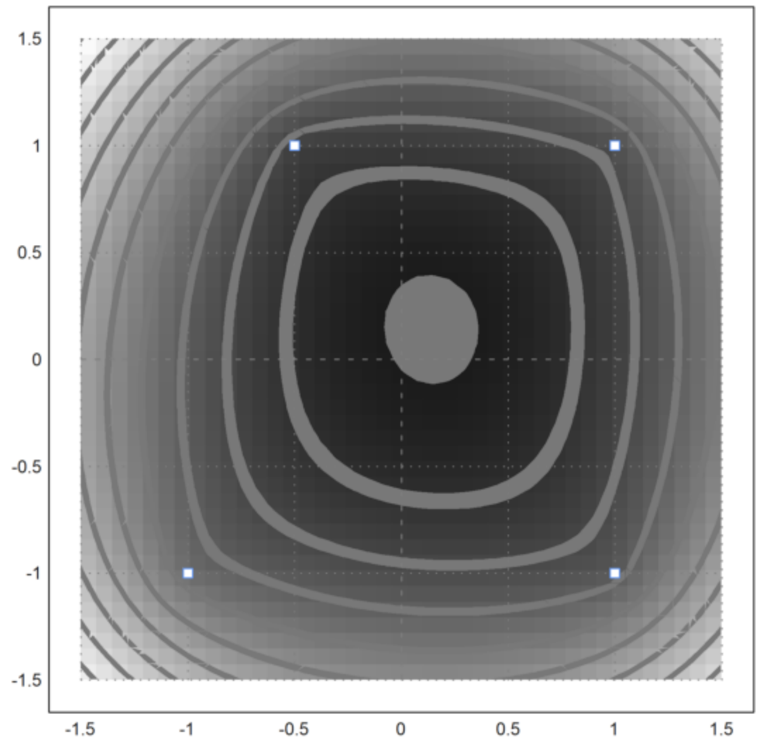
Empat poin

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```




```
>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 ada minimum dan tidak tercapai di salah satu simpul A, B, C atau D:

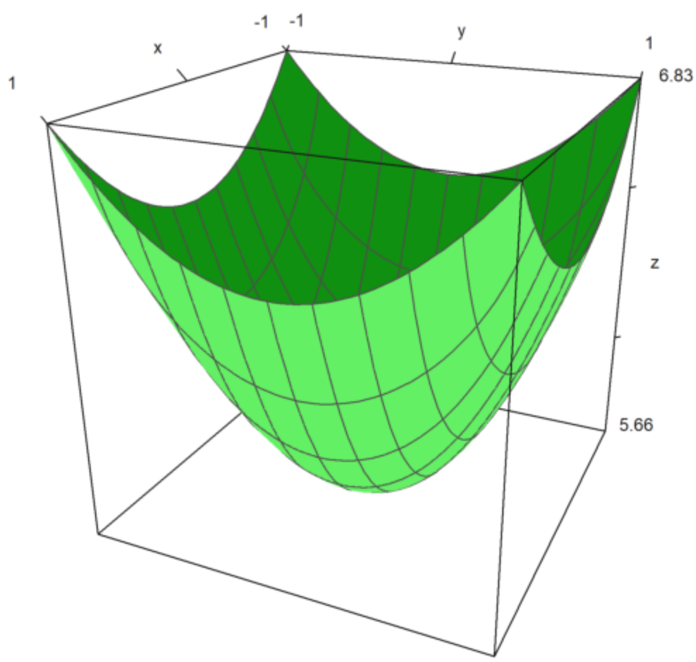
```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
[0.142858,  0.142857]
```

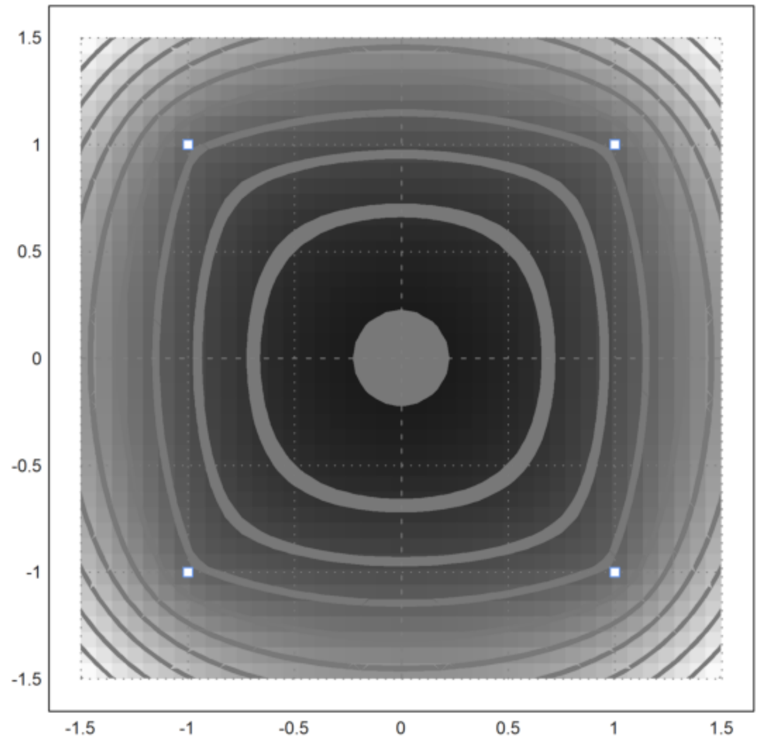
Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kami berharap bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```




```
>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 membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometri.e dari Euler untuk ini.

```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[- a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

[- a, - 1, 0]

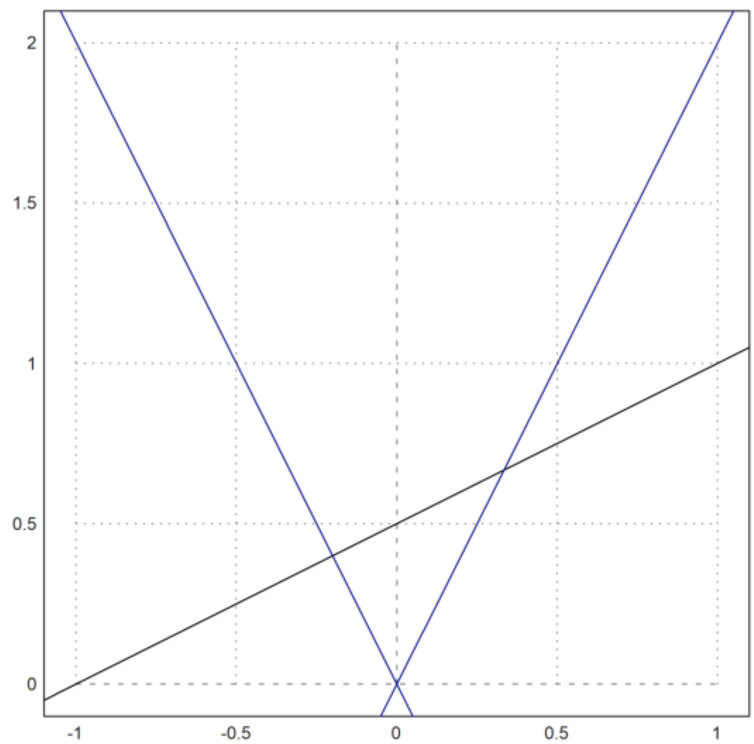
Kemudian saya baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

[- 1, 2, 1]

Kami merencanakan semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
>color(black); plotLine(g(),"")
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```



Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[0, u]
```

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2\,u}{a^2+1}-u\right)^2+\frac{a^2\,u^2}{\left(a^2+1\right)^2}}$$

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u+2}{5}-u\right)^2+\frac{\left(2\,u-1\right)^2}{25}}$$

Dan temukan pusat kedua lingkaran yang jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u=\frac{-\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1},u=\frac{\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1}\right]$$

Ada dua solusi.

Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()
```

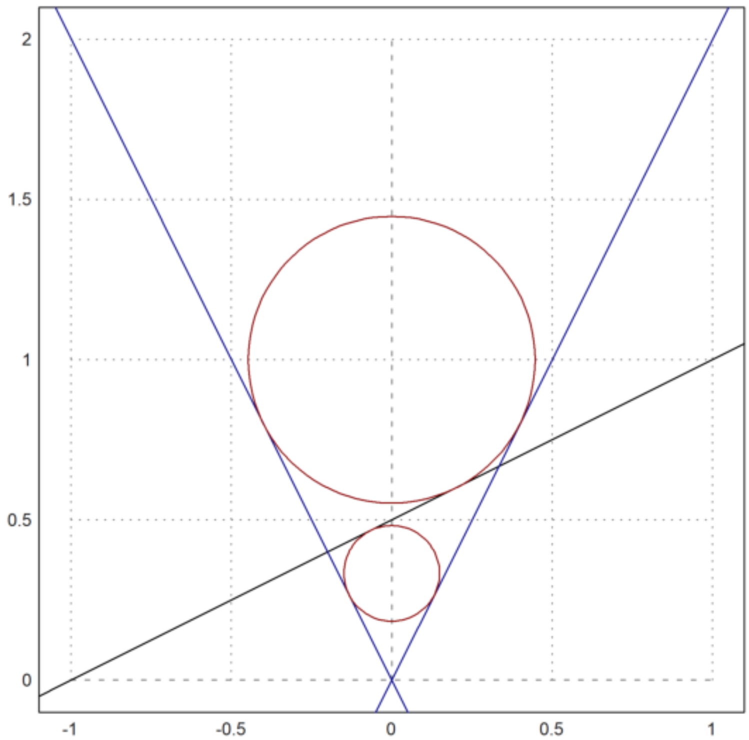
```
[0.333333,  1]
```

```
>dd := d()
```

```
[0.149071,  0.447214]
```

Plot lingkaran ke dalam gambar.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
>insimg;
```



Plot dengan Povray

Selanjutnya kami merencanakan semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.

Pertama kita memuat fungsi povray.

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

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

Kami mengatur adegan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya kita menulis dua bidang ke file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan bidang terbatas pada kerucut.

```
>gp=g();
>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 menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian kami menghasilkan dua titik di mana bola menyentuh bidang. Ini adalah fokus dari elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow)));
```

Selanjutnya kita hitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
```

Kami 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 menghasilkan pita abu-abu, di mana bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-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();
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Untuk mendapatkan Anaglyph ini kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali kemudian.

```
>function scene () ...
```

```
global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

Anda membutuhkan kacamata merah/sian untuk menghargai efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

```
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povanaglyph:
    povray(currentfile,w,h,aspect,exit);
```

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file ”spherical.e” di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569,  1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi spherical).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467’ E 110°23.050’
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333’ E 110°49.683’
S 6°59.050’ E 110°24.533’
```

Pertama kita menghitung vektor dari satu ke yang lain pada bola ideal. Vektor ini [pos,jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7°.

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20’26.60’’
53.8945384608
```

Ini adalah perkiraan yang baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km", // perkiraan jarak FMIPA-Semarang
```

```
88.0114026318 km
```

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
     65.34°
```

Sudut segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo); degprint(asu
```

```
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
```

```
2116.02948749 km^2
```

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

```
2123.64310526 km^2
```

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor sadd.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
>sposprint(Tugu), sposprint(Monas)
```

```
S 7°46.998' E 110°21.966'
S 6°10.500' E 106°48.717'
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

```
>esdist(Tugu,Monas)->" km", // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
431.565659488 km
```

Judulnya sama dengan judul yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17'2.85''
```

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
S 6°10.500' E 106°48.717'
```

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

```
S 6°10.500' E 106°48.717'
```

Tentu kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh.

```
>sposprint(p), 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°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang 30°, melainkan jalur terpendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
79.69°
```

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1;  dist=esdist(P1,P2); ...
>  loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°
81.67°
83.71°
85.78°
87.89°
90.00°
92.12°
94.22°
96.29°
98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
>  loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmprint(esdist(p,Monas))
```

```
0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
>  loop 1 to rows(v); sposprint(v[#]), end;
```

```
S 7°46.998' E 110°21.966'
S 7°37.422' E 110°0.573'
S 7°27.829' E 109°39.196'
S 7°18.219' E 109°17.834'
S 7°8.592' E 108°56.488'
S 6°58.948' E 108°35.157'
S 6°49.289' E 108°13.841'
S 6°39.614' E 107°52.539'
S 6°29.924' E 107°31.251'
S 6°20.219' E 107°9.977'
S 6°10.500' E 106°48.717'
```

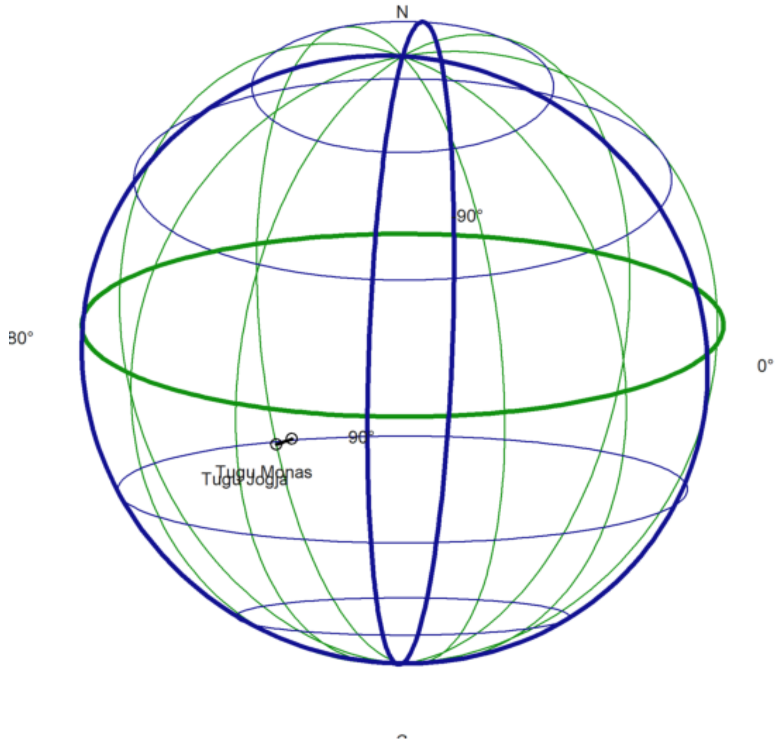
Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction
```

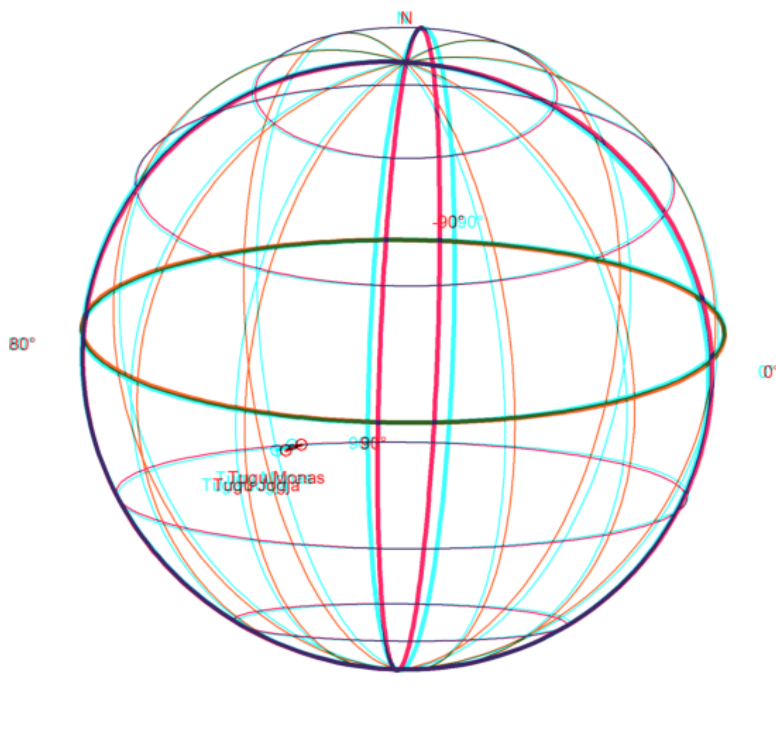
Sekarang rencanakan semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/sian.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



Geometri Simbolik

```
>A &= [2,0]; B &= [0,2]; C &= [3,3]; // menentukan tiga titik A, B, C
>c &= lineThrough(B,C) // c=BC
```

```
[- 1, 3, 6]
```

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$3\,y-x=6$$

$$\left[y=\frac{x}{3}+2\right]$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

```
[3, 1, 6]
```

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

```
 6  12
[-, --]
 5  5
```

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[\frac{6}{5},\frac{12}{5}\right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{2^{\frac{5}{2}}}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{4}, \frac{7}{4}, \frac{5}{2^{\frac{3}{2}}}\right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{2^{\frac{3}{2}}}$$

$$1.767766952966368$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{3}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2
```

$$\left[\frac{\sqrt{2}\,\sqrt{10}+2}{4}, \frac{\sqrt{2}\,\sqrt{10}+2}{4}\right]$$

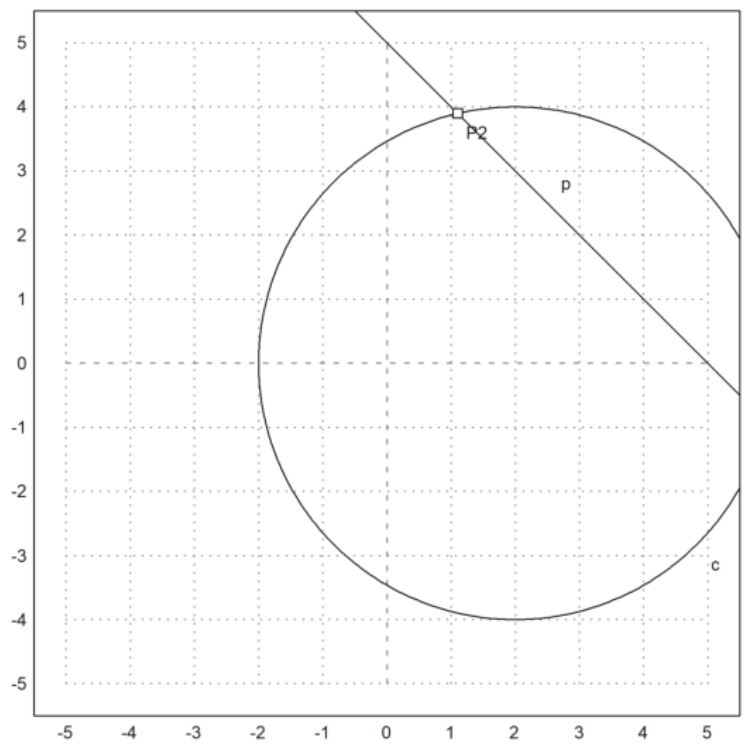
```
>P() //
```

```
[1.61803,  1.61803]
```

```
>A &:= [2,0]; c=circleWithCenter(A,4);
>B &:= [2,3]; C &:= [3,2]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2,
```

```
[5.89792,  -0.897916]
[1.10208,  3.89792]
```

```
>plotPoint(P1); plotPoint(P2):
```




```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

$$[2, 0, 4]$$

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

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

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[\left[\frac{\sqrt{23}+7}{2},\frac{3-\sqrt{23}}{2}\right],\left[\frac{7-\sqrt{23}}{2},\frac{\sqrt{23}+3}{2}\right]\right]$$

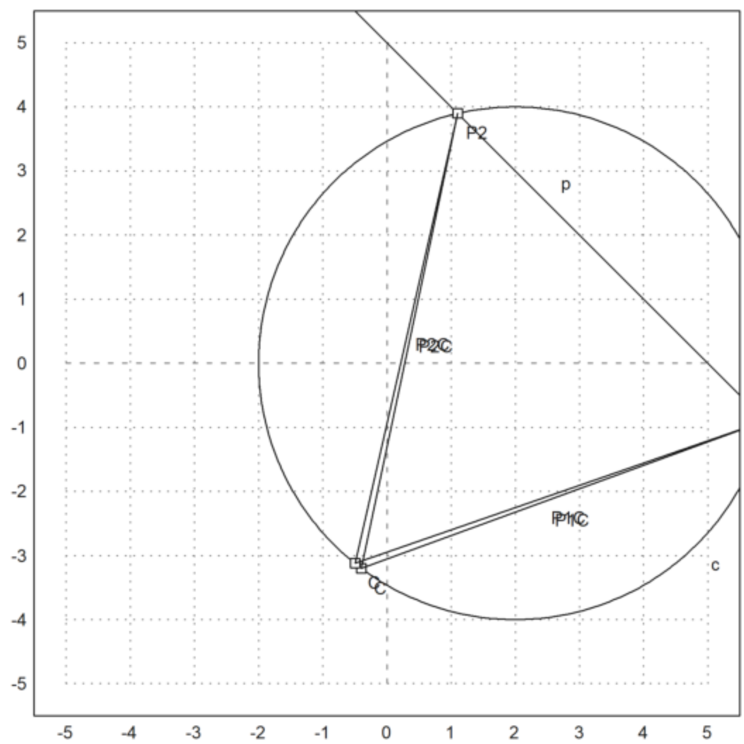
```
>C=A+normalize([-3,-4])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
57°58'20.06''
```

```
>C=A+normalize([-4,-5])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

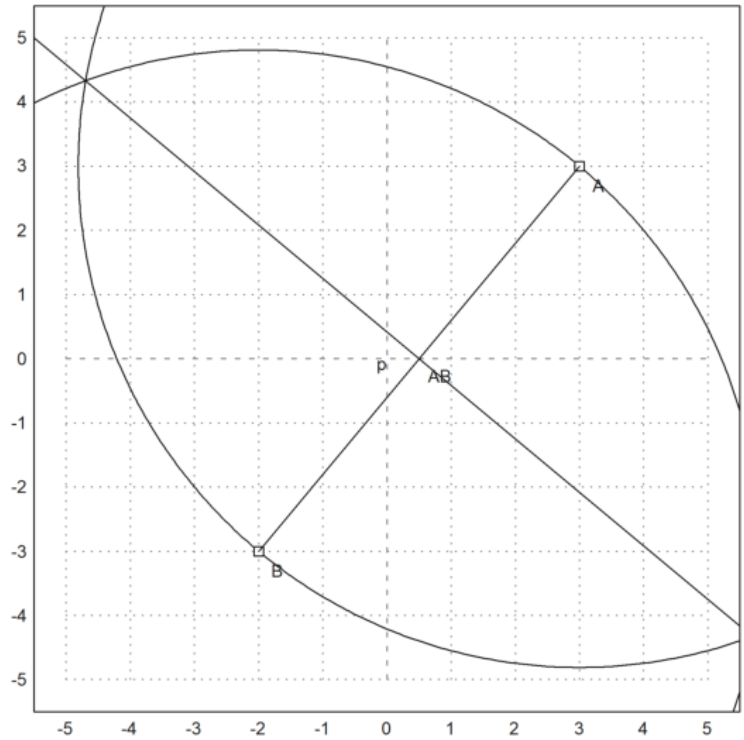
```
57°58'20.06''
```

```
>insimg;
```



Garis Sumbu

```
>A=[3,3]; B=[-2,-3];
>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):
```




```
>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];
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

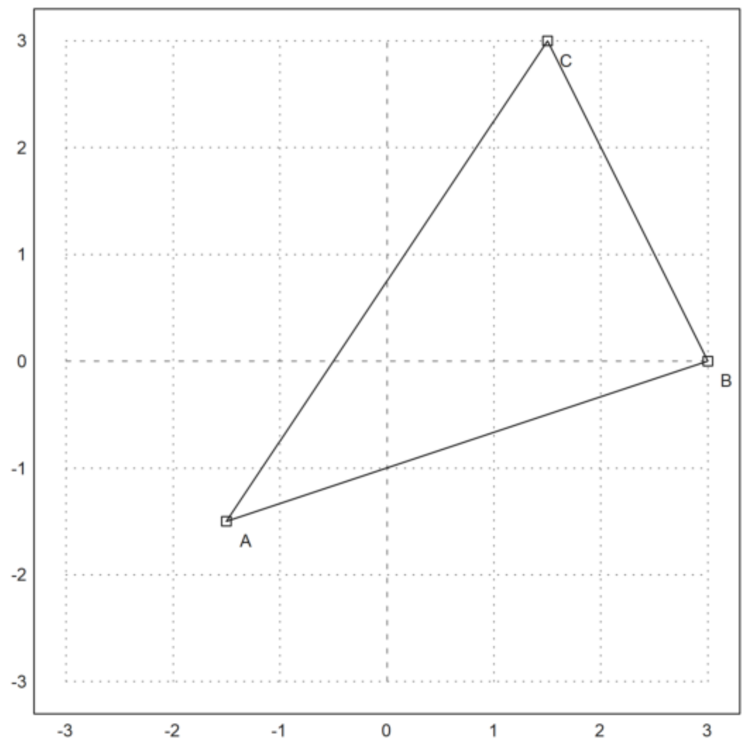
$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y=\frac{\left(b_2-a_2\right)\,x-a_1\,b_2+a_2\,b_1}{b_1-a_1}\right]$$

```
>A::=[-1.5,-1.5]; B::=[3,0]; C::=[1.5,3];
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
```




```
>$areaTriangle(A,B,C)
```

$$-\frac{63}{8}$$

```
>c &= lineThrough(A,B)
```

```
   3  9    9
[- -, -, - -]
   2  2    2
```

```
>$getLineEquation(c,x,y)
```

$$\frac{9\,y}{2}-\frac{3\,x}{2}=-\frac{9}{2}$$

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{\sqrt{2}\,\left(\frac{9\,y}{2}-\frac{3\,x}{2}+\frac{9}{2}\right)}{3\,\sqrt{5}}$$

$$\frac{21}{2^{\frac{3}{2}}\,\sqrt{5}}$$

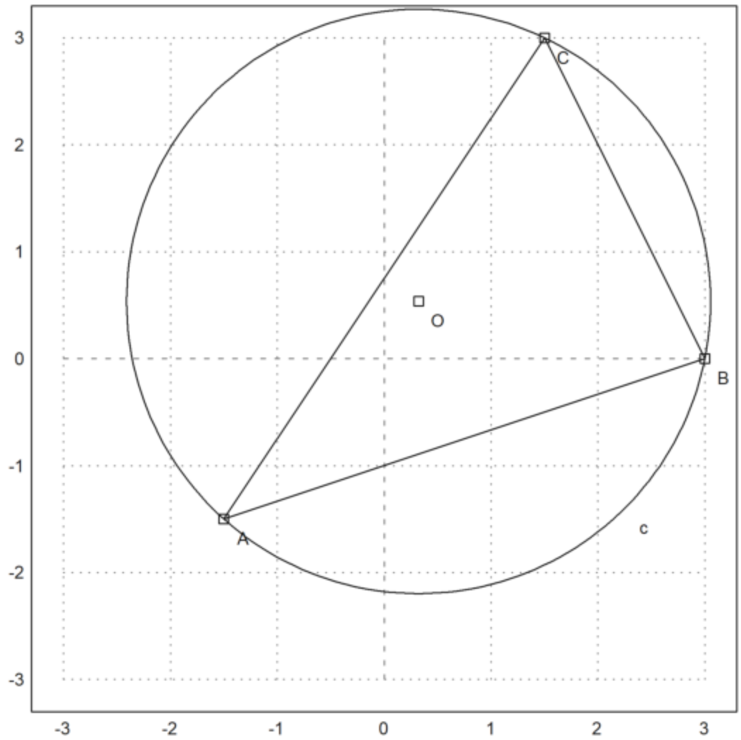
```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

$$\left(y-\frac{15}{28}\right)^2+\left(x-\frac{9}{28}\right)^2=\frac{2925}{392}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{9}{28},\frac{15}{28}\right]$$

```
>plotCircle(LL()); plotPoint(O(),"O"):
```



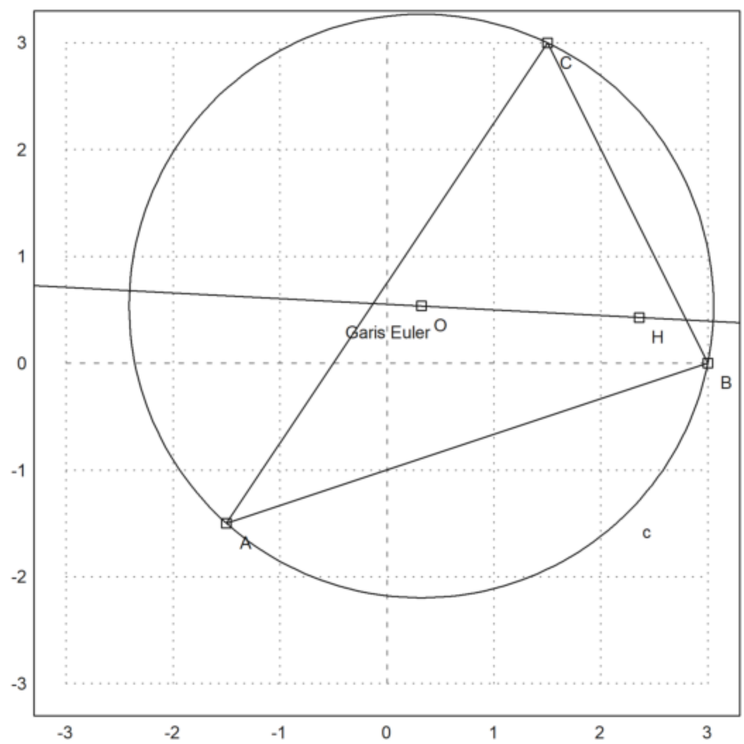

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
>  perpendicular(B,lineThrough(A,C))); $H
```

$$\left[\frac{33}{14},\frac{3}{7}\right]$$

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{57\,y}{28}-\frac{3\,x}{28}=-\frac{9}{8}$$

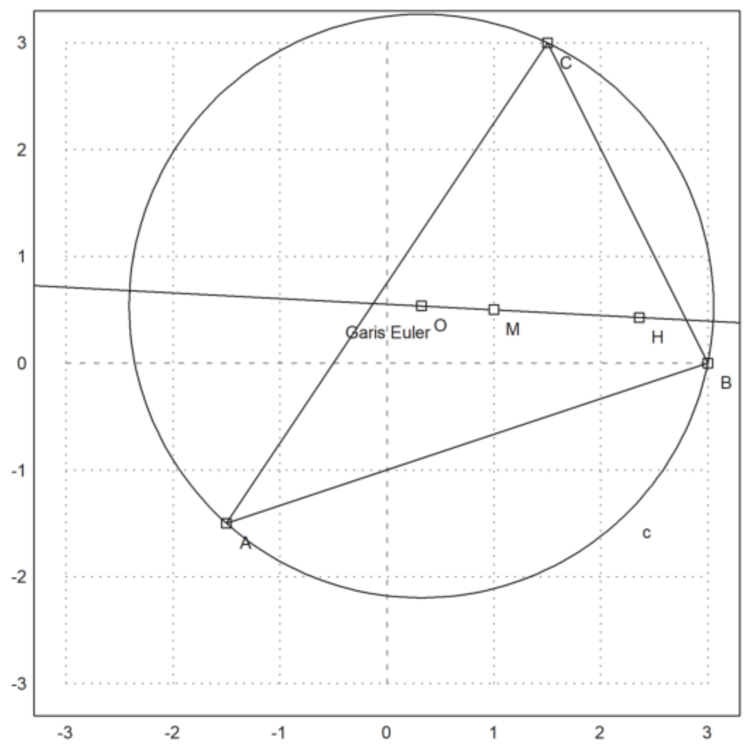
```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```




```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{9}{8}=-\frac{9}{8}$$

```
>plotPoint(M(),"M"): // titik berat
```




```
>$distance(M,H)/distance(M,O)|radcan
```

$$2$$

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\,\sqrt{13}}\right)$$

```
60°15'18.43''
```

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

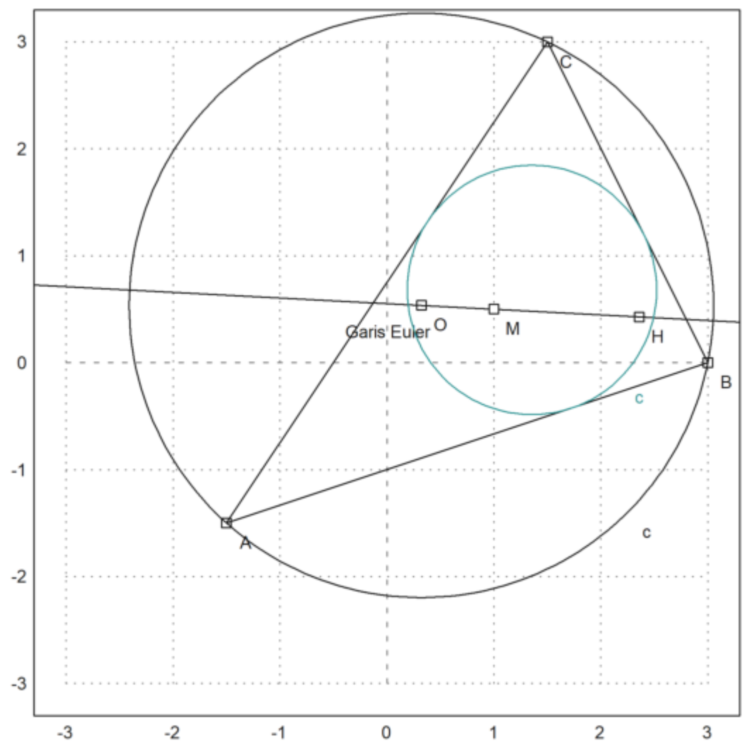
$$\left[\frac{\left(3\,2^{\frac{3}{2}}+3\right)\,\sqrt{5}\,\sqrt{13}-45\,\sqrt{2}+9}{28},\frac{\left(3\,\sqrt{2}-9\right)\,\sqrt{5}\,\sqrt{13}+15\,2^{\frac{3}{2}}+15}{28}\right]$$

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{\left(-369\,\sqrt{2}-279\right)\,\sqrt{5}\,\sqrt{13}+1035\,\sqrt{2}+5526}}{7\,2^{\frac{3}{2}}}$$

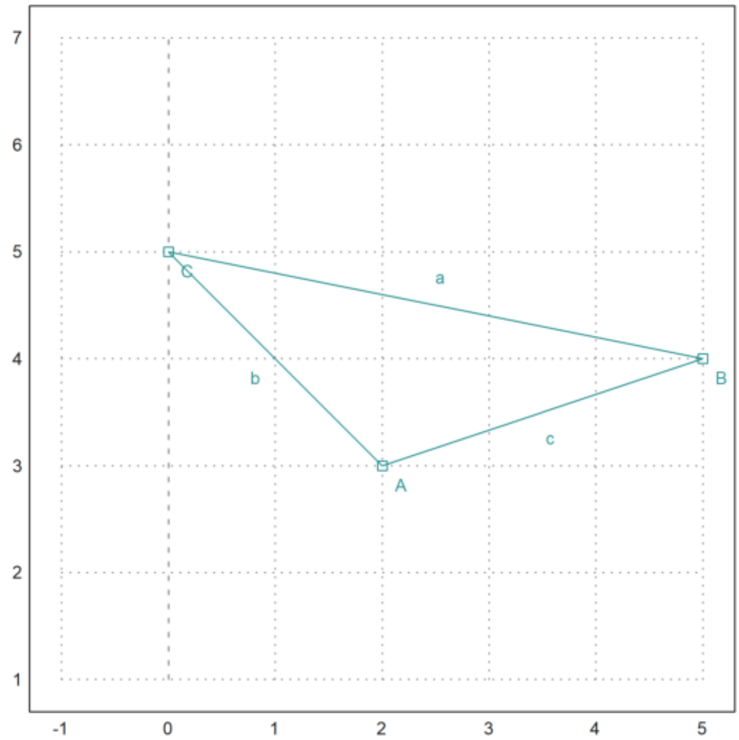
```
>LD &=  circleWithCenter(Q,r); // Lingkaran dalam
```

```
>color(5); plotCircle(LD()):
```



contoh lain dari materi trigonometri rasional

```
>A&:=[2,3]; B&:=[5,4]; C&:=[0,5]; ...
>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;
```




```
>$distance(A,B)
```

$$\sqrt{10}$$

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$10$$

$$8$$

$$26$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{14}{\sqrt{10}\,\sqrt{26}}\right)$$

```
29.7448812969
```

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

$$64 = 320\ (1 - x)$$

$$\left[x = \frac{4}{5}\right]$$

```
>sb &= spread(b,a,c); $sb
```

$$\frac{16}{65}$$

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{16}{65}$$

```
>ha &= c*sb; $ha
```

$$\frac{32}{13}$$

```
>$sqrt(ha)
```

$$\frac{2^{\frac{5}{2}}}{\sqrt{13}}$$

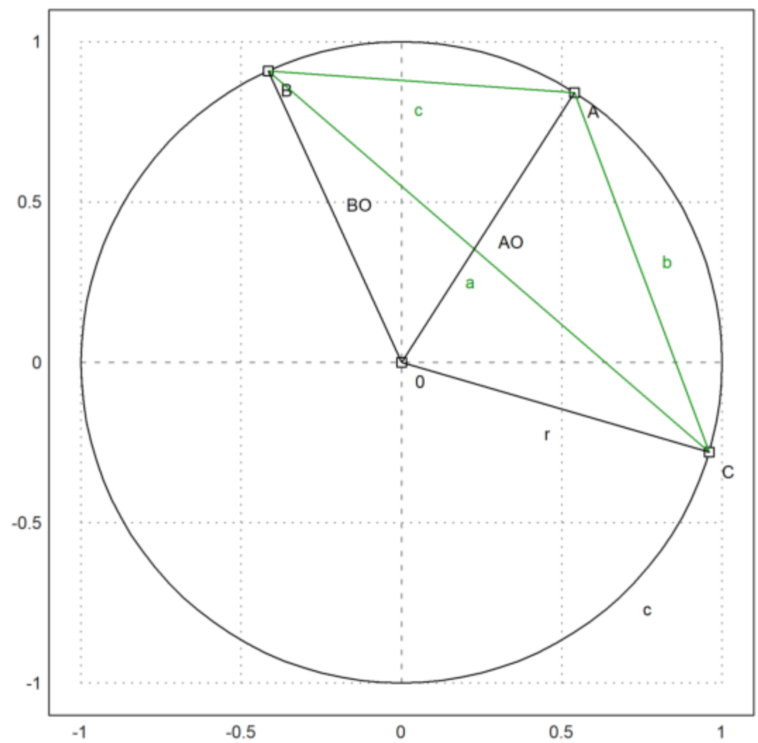
```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{2^{\frac{3}{2}}\ \sqrt{26}}{\sqrt{13}}$$

```
>$areaTriangle(B,A,C)
```

$$4$$

```
>setPlotRange(1); ...
>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;
```




```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

$$-\frac{a\,b\,c}{c^2-2\,b\,c+a\,\left(-2\,c-2\,b\right)+b^2+a^2}$$

```
>function periradius(a,b,c) &= rabc;
```

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

```
>periradius(a,b,c)
```

```
1
```

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

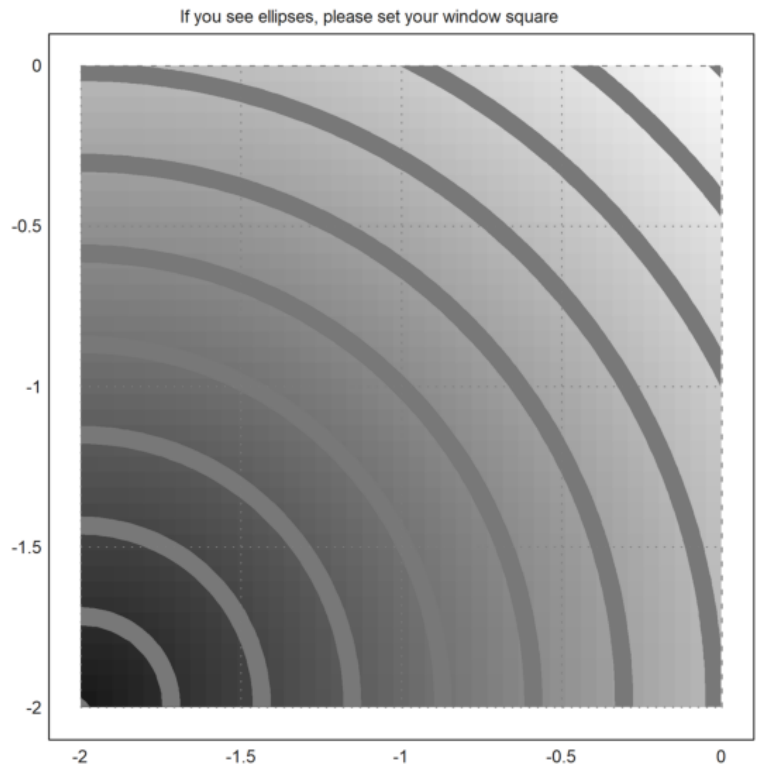
$$0$$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

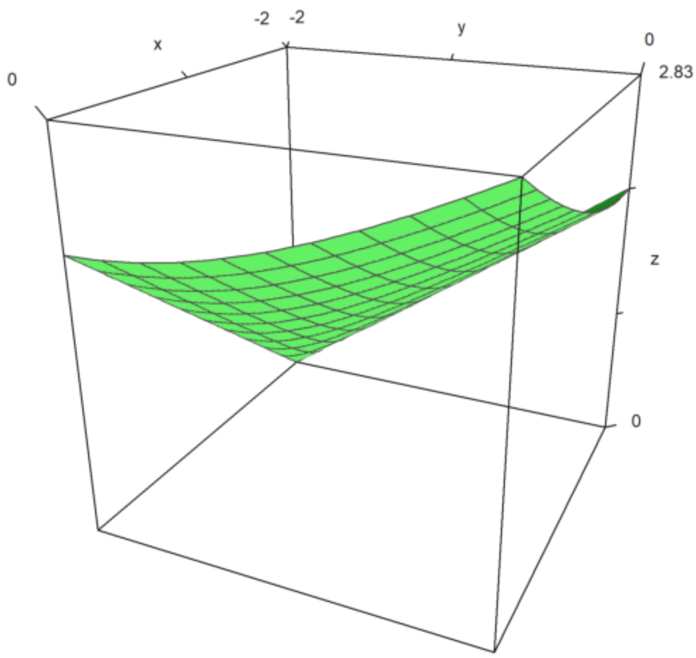
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

```
>&remvalue();
>A=[-2,-2];
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
>title="If you see ellipses, please set your window square"):
```



dan grafiknya juga agak sederhana: bagian atas kerucut:

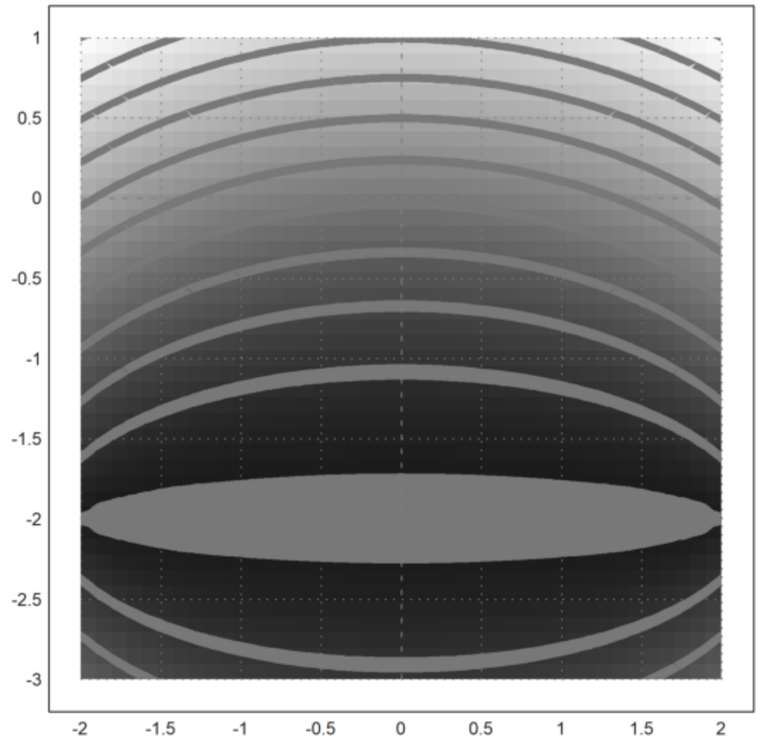
```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```



Ternyata setelah mencoba yang bisa hanya dengan memasukkan angka 1, karena ketika memakai angka 2, plot tidak membentuk kerucut diatas.

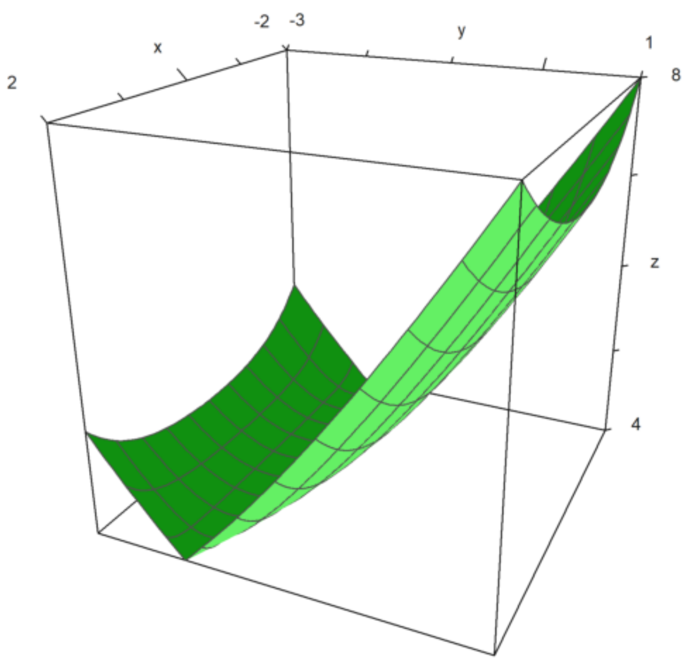
Dua poin

```
>B=[2,-2];
>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):
```



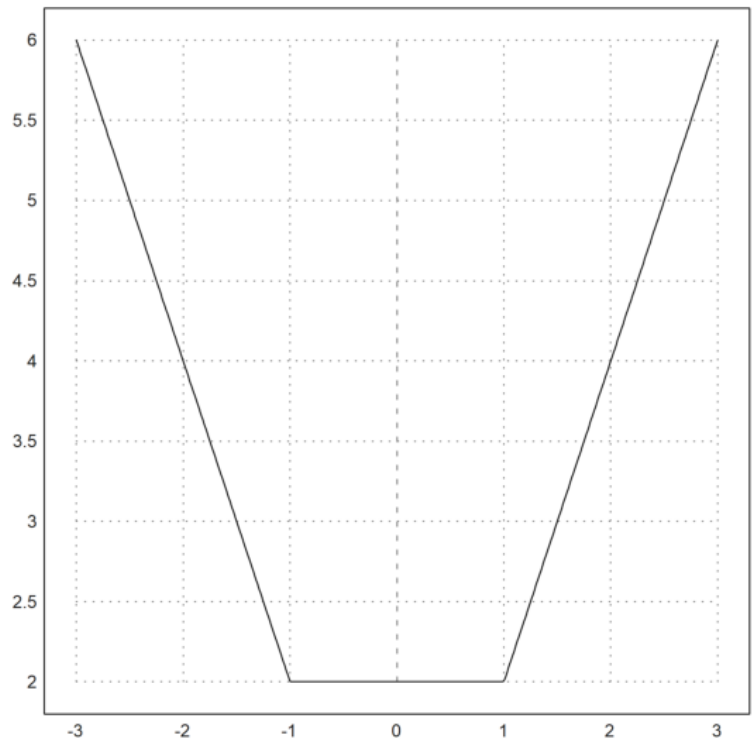
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan garis (AB) lebih terkenal:

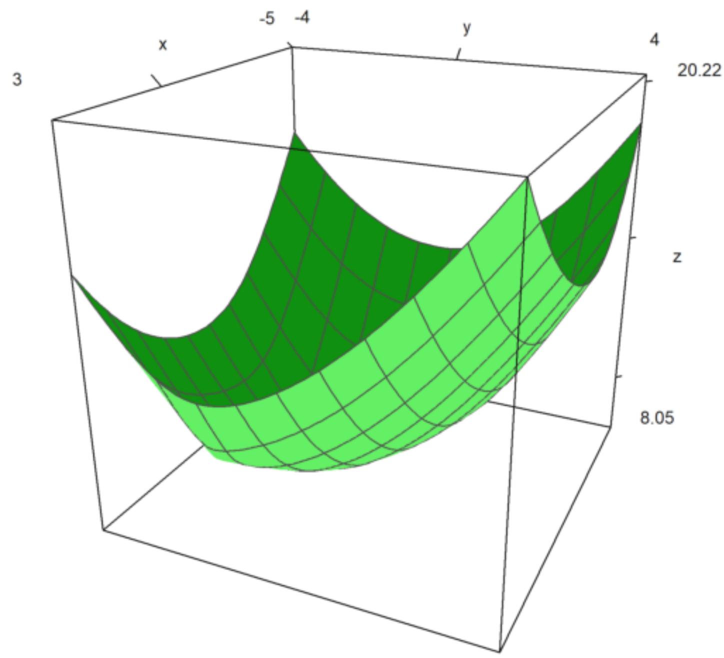
```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



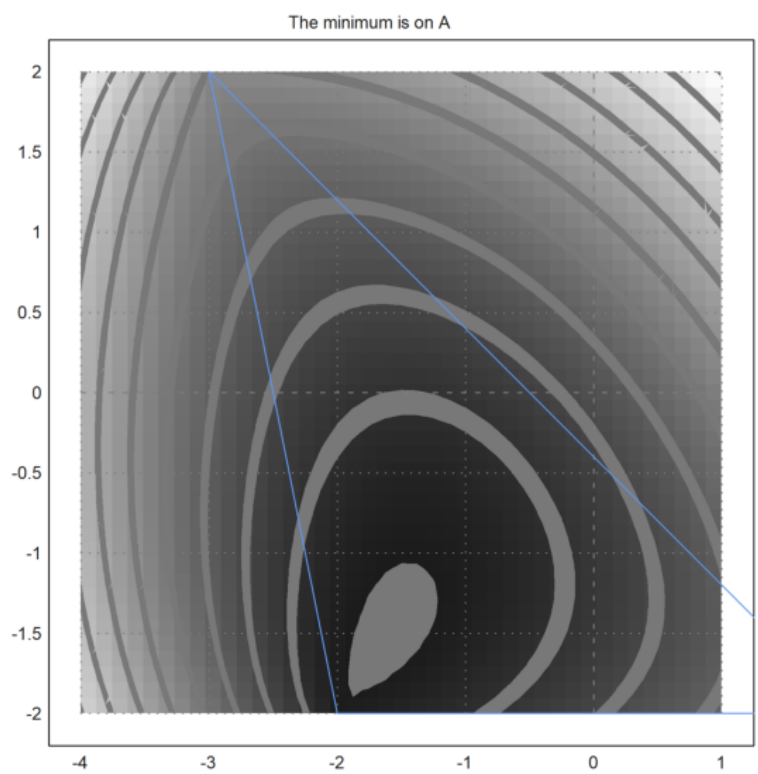
Tiga poin

Contoh:

```
>C=[-3,2];
>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;
```

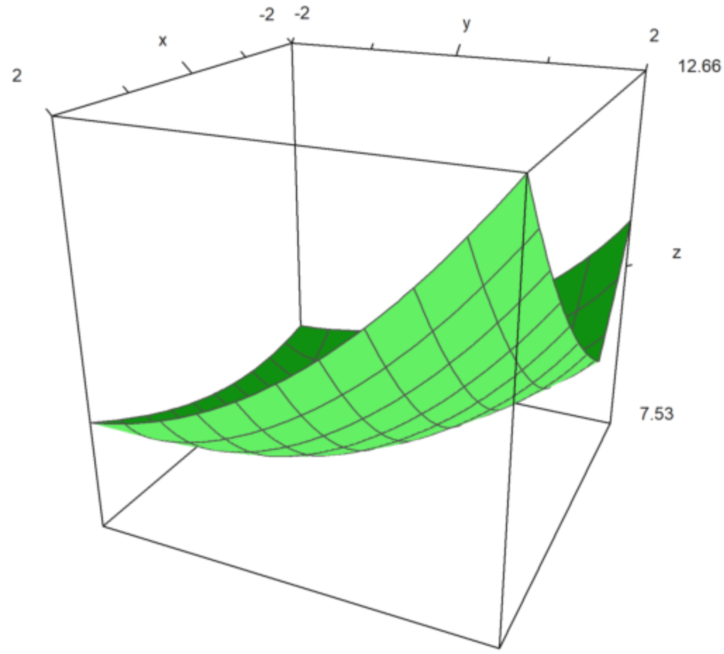



```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

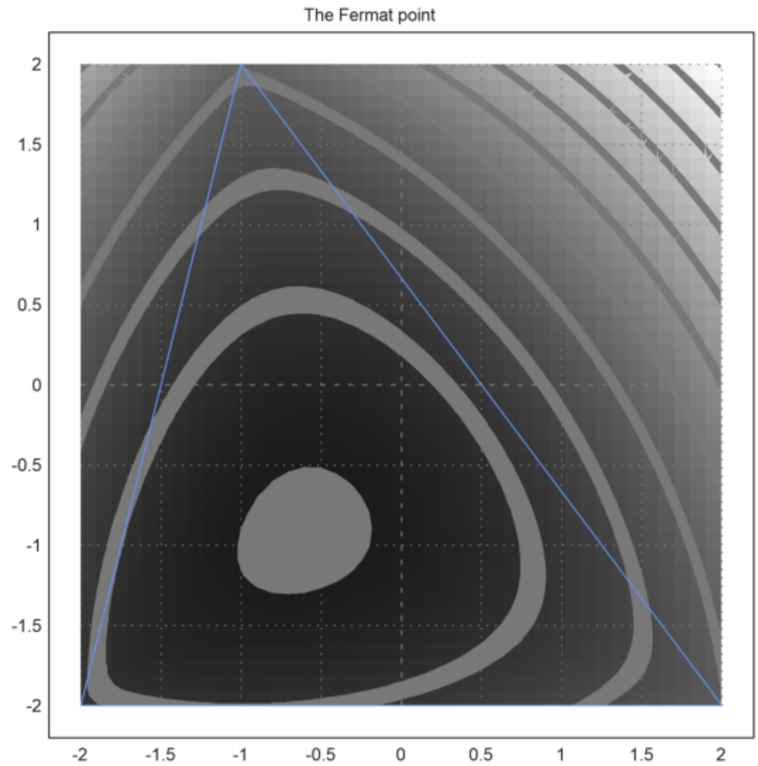


Tetapi jika semua sudut segitiga ABC kurang dari 120 °, minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing 120 °):

```
>C=[-1,2];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```



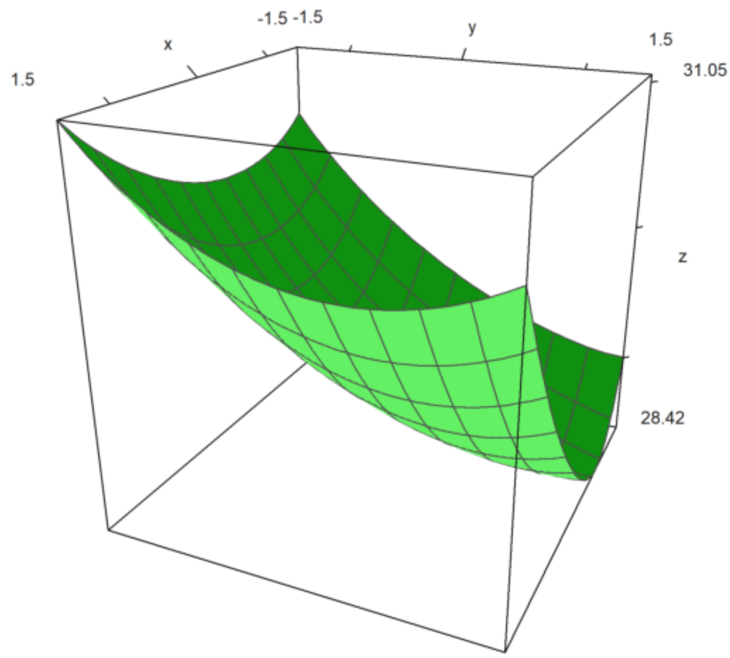

```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



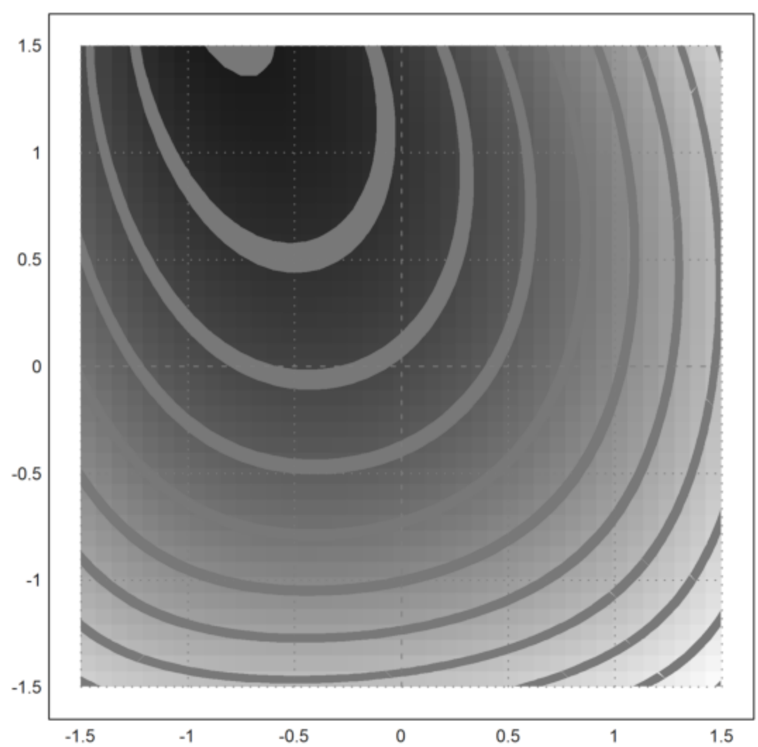
Empat poin

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[2,21];
>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;
```



```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[2,a])
```

```
[- a, 2, 0]
```

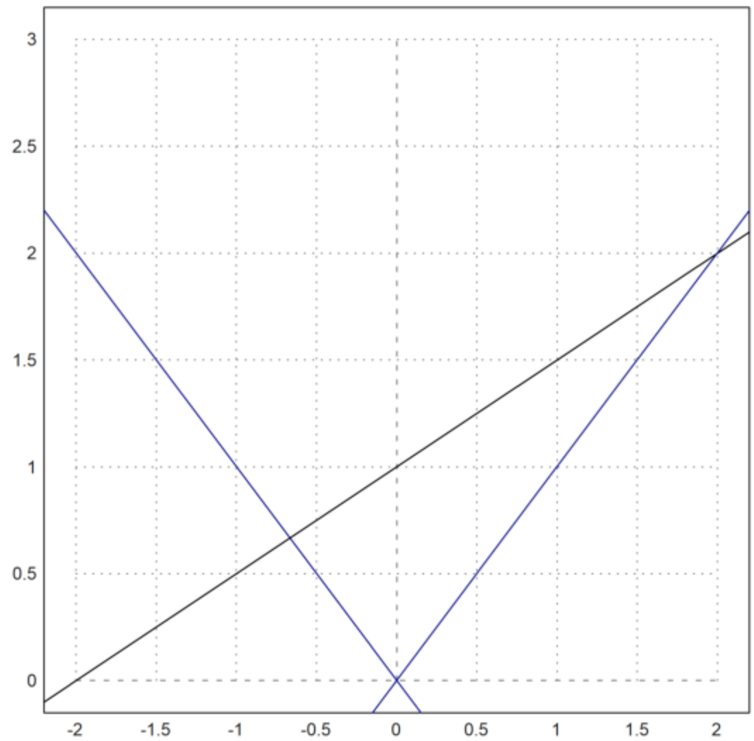
```
>g2 &= lineThrough([0,0],[-2,a])
```

```
[- a, - 2, 0]
```

```
>g &= lineThrough([-2,0],[2,2])
```

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

```
>setPlotRange(-2,2,0,3);
>color(black); plotLine(g(),"")
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```



Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[0, u]
```

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2\,u}{a^2+4}-u\right)^2+\frac{4\,a^2\,u^2}{\left(a^2+4\right)^2}}$$

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u+4}{5}-u\right)^2+\frac{\left(2\,u-2\right)^2}{25}}$$

Dan temukan pusat kedua lingkaran yang jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u=\frac{-\sqrt{5}\,\sqrt{a^2+4}+a^2+4}{a^2-1},u=\frac{\sqrt{5}\,\sqrt{a^2+4}+a^2+4}{a^2-1}\right]$$

Ada dua solusi.

```
>u := sol()
```

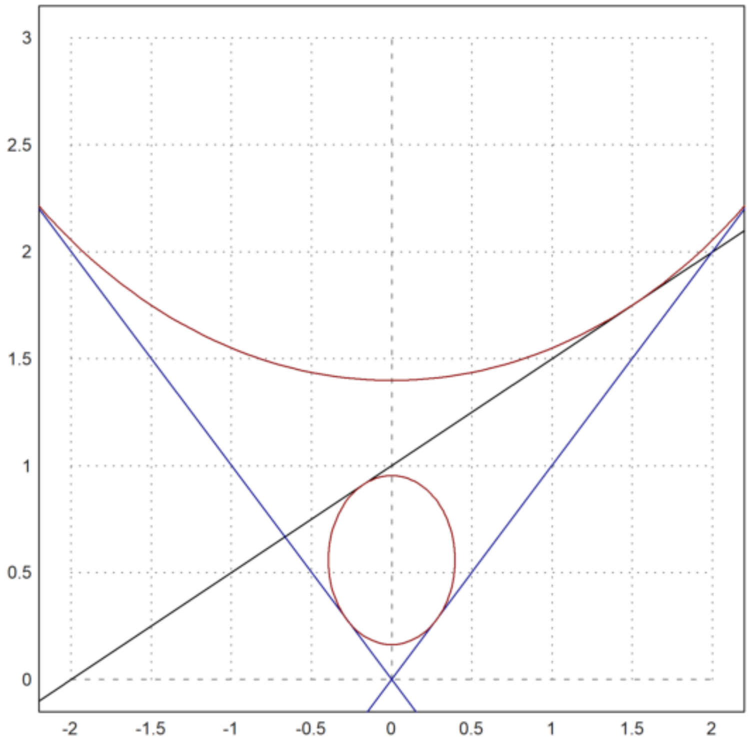
```
[0.558482,  4.77485]
```

```
>dd := d()
```

```
[0.394906,  3.37633]
```

Plot lingkaran ke dalam gambar.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
>insimg;
```



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:

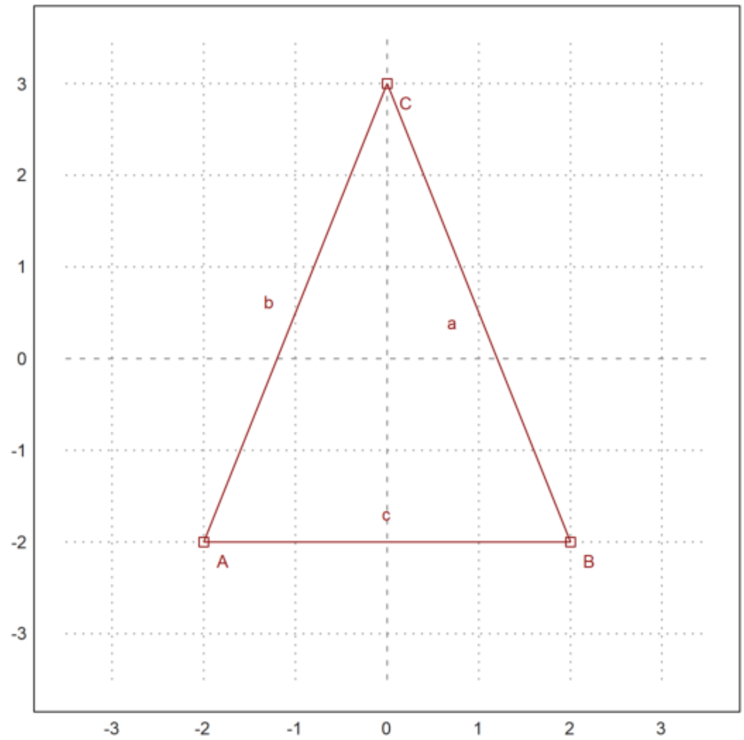
- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah (360/n).
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan (360/n).
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- 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.

Penyelesaian :

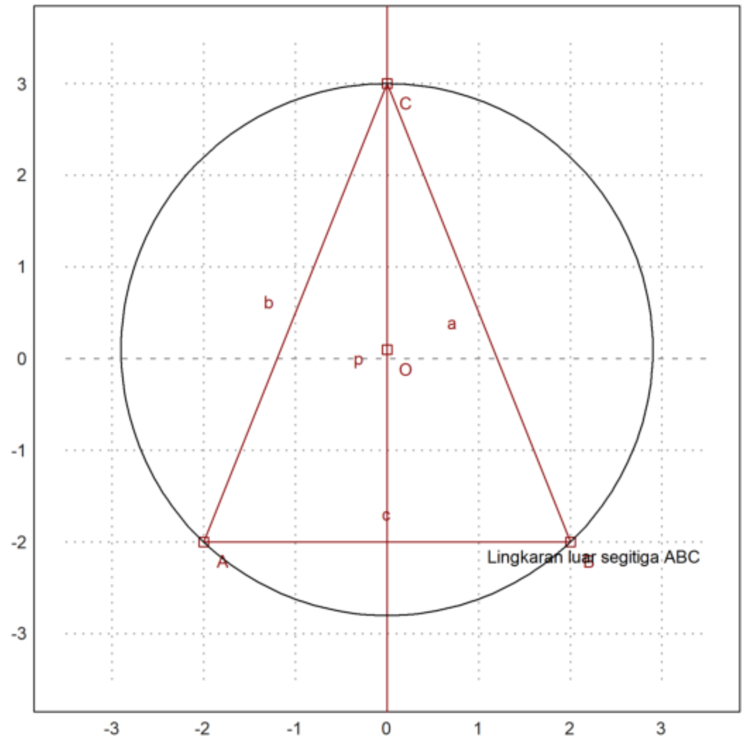
```
>load geometry
```

```
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"):
```

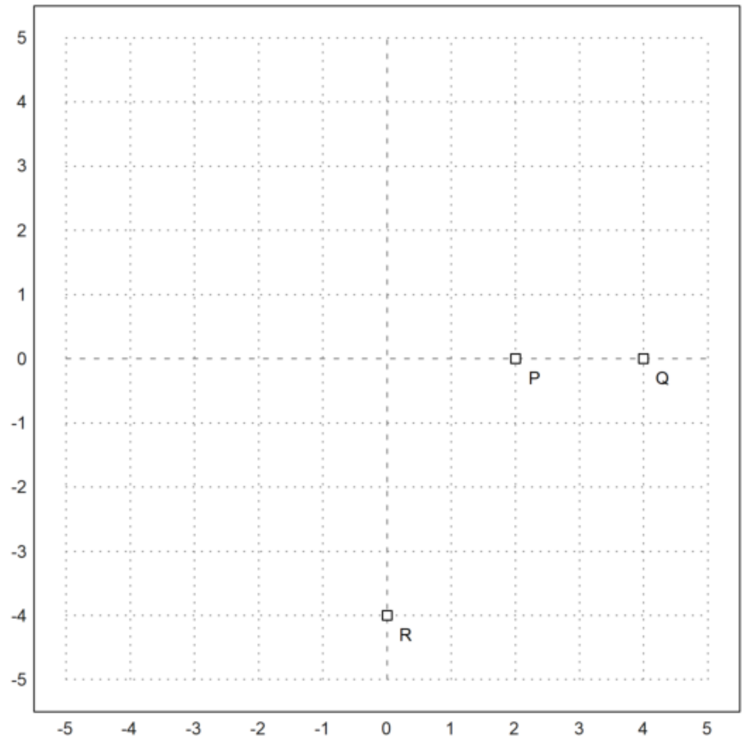


2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:
- Misalkan persamaan parabolanya y= axˆ2+bx+c.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

Penyelesaian :

```
>load geometry;
>setPlotRange(5); P=[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])
```

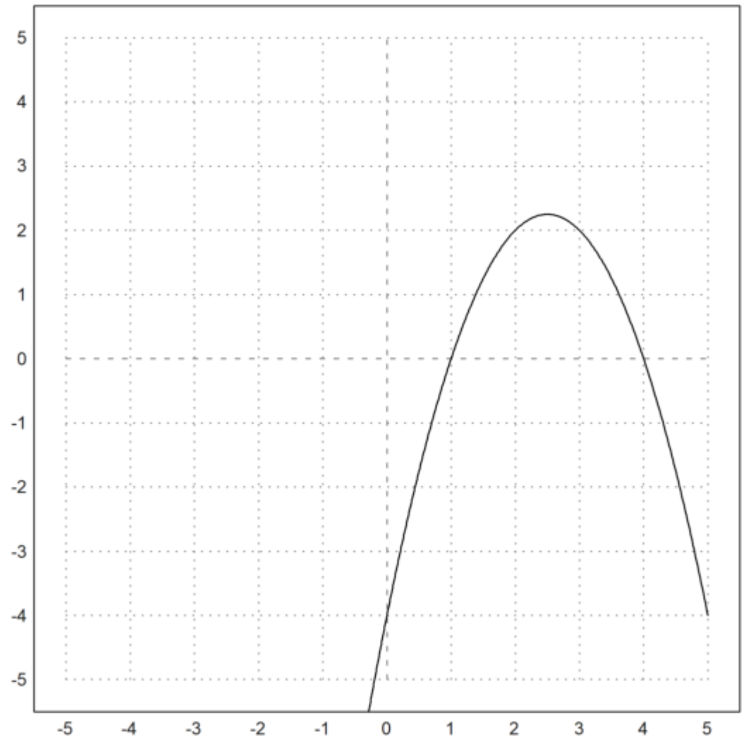
```
[[a = - 1, b = 5, c = - 4]]
```

Sehingga didapatkan nilai a = -1, b = 5 dan 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
```

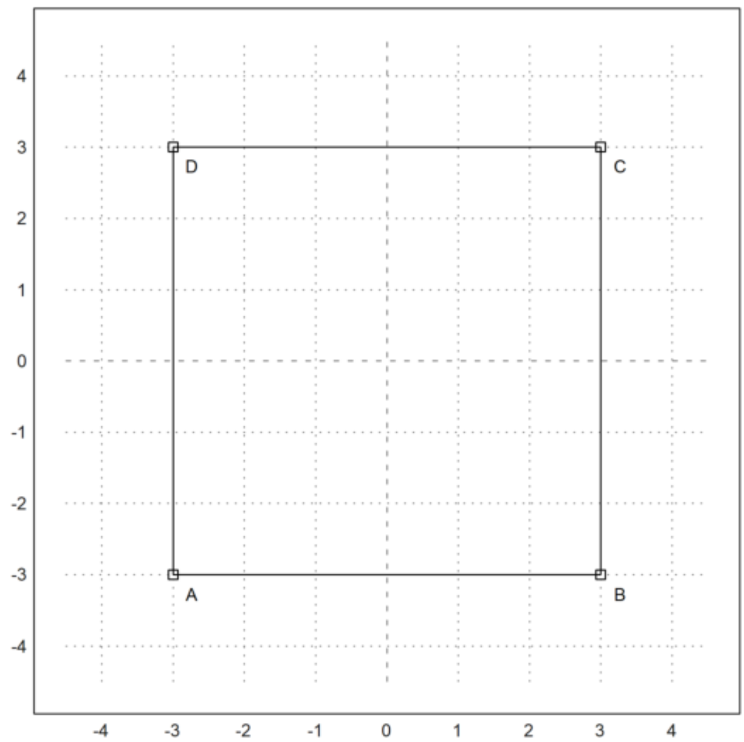
singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

Penyelesaian :

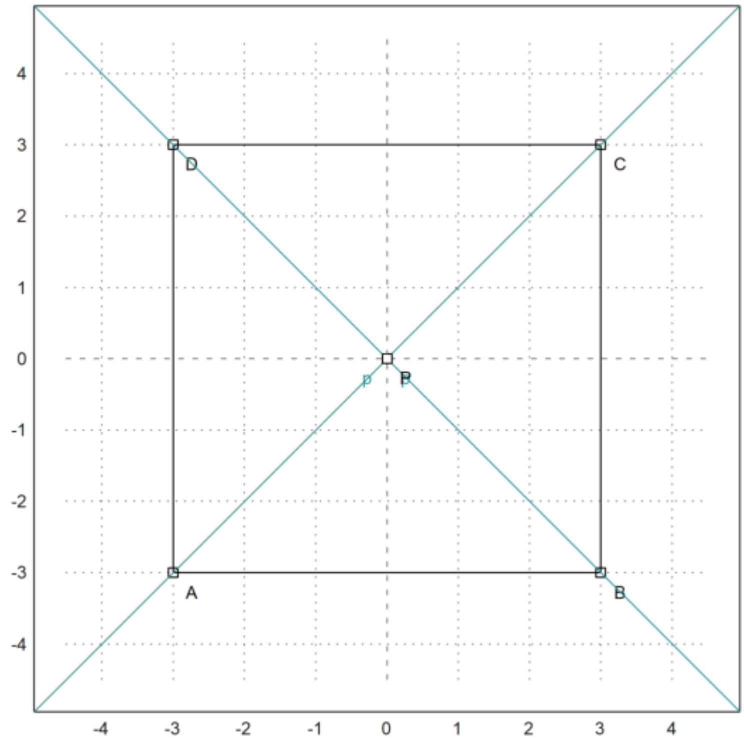
```
>load geometry
```

```
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):
```

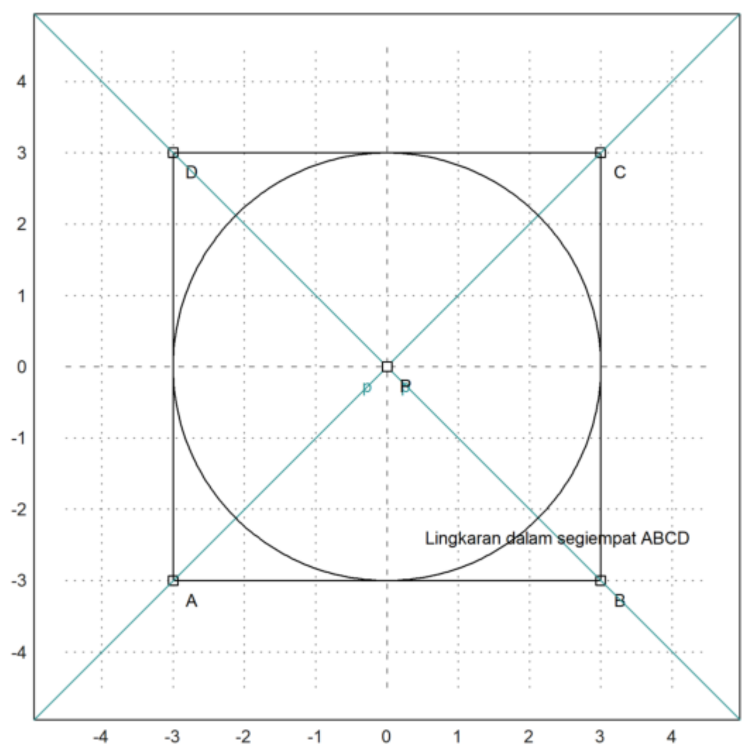



```
>l=angleBisector(A,B,C);
>m=angleBisector(B,C,D);
>P=lineIntersection(l,m);
>color(5); plotLine(l); plotLine(m); color(1);
>plotPoint(P,"P"):
```



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
```

36

```
>AD.BC
```

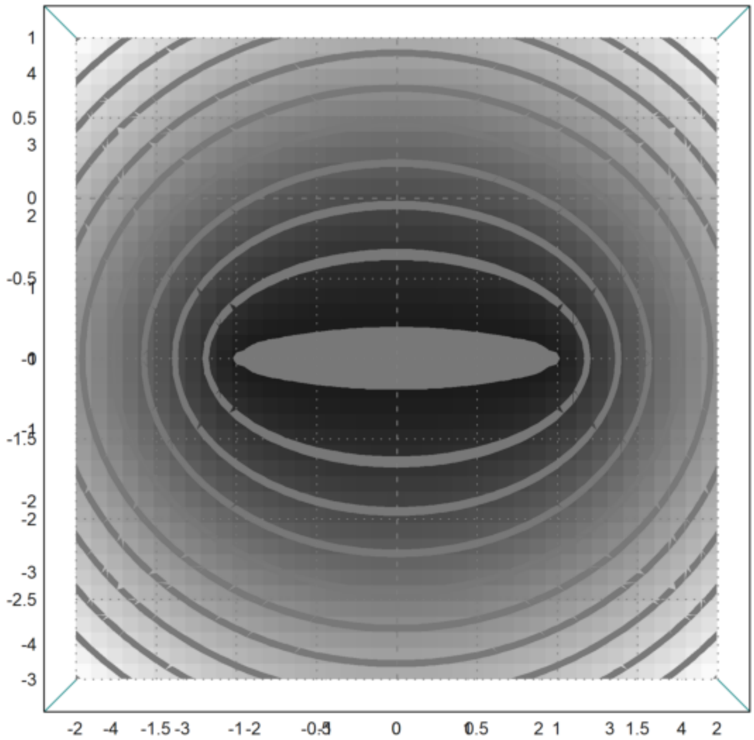
36

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).

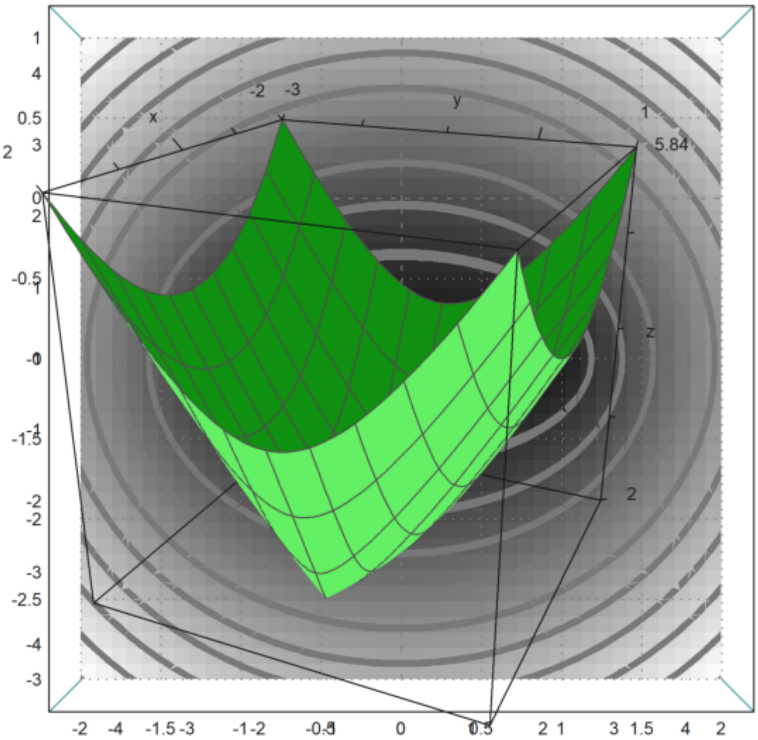
Penyelesaian :
Diketahui kedua titik fokus P = [-1,-1] dan Q = [1,-1]

```
>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):
```



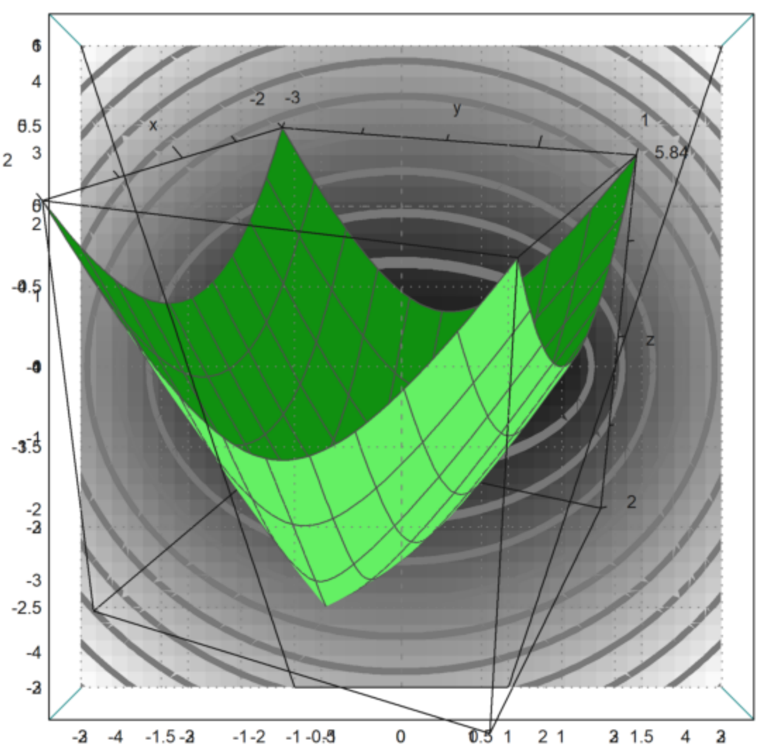
Grafik yang lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Batasan ke garis PQ

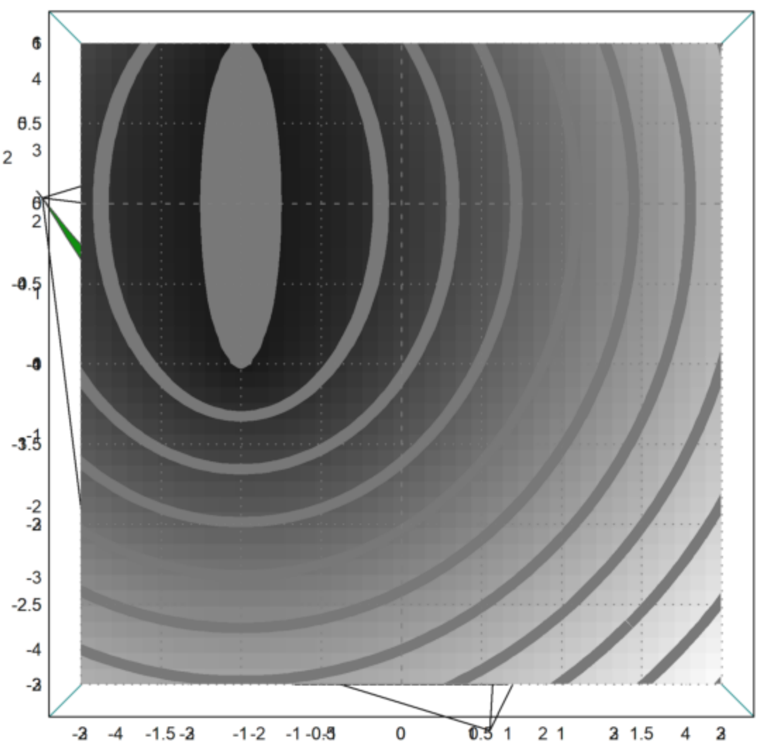
```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

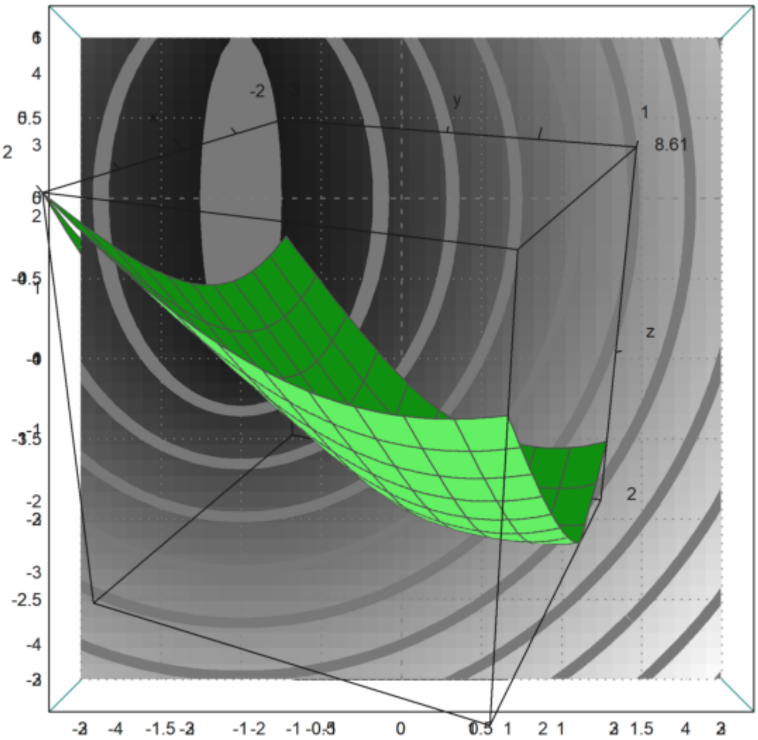
Penyelesaian :

```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-p[1])^2+(y-p[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x+Q[1])^2+(y+Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```

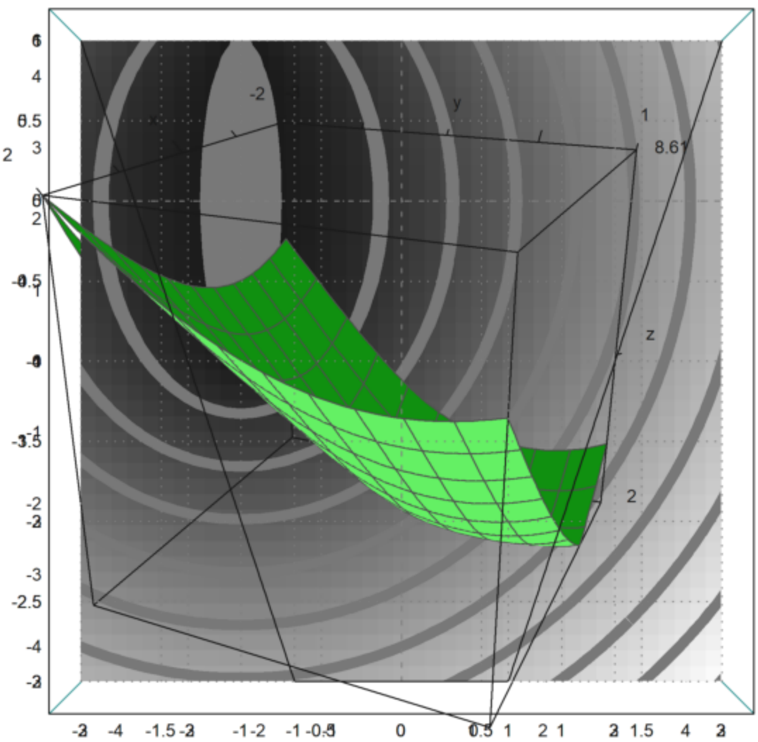


Grafik yang lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```




```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



BAB 6

KB PEKAN 10; MENGGUNAKAN EMT UNTUK STATISTIKA

[a4paper,10pt]article eumat

Nama : Nafisatul Iqima
NIM : 22305144037
Kelas : Matematika E 2022

Array adalah kumpulan-kumpulan variabel yang menyimpan data dengan tipe yang sama atau data-data yang tersusun secara linear dimana di dalamnya terdapat elemen dengan tipe yang sama.

Vektor digunakan untuk menggambarkan array angka satu dimensi. Vektor memiliki panjang, yang merupakan jumlah elemen dalam array.

Sedangkan matriks digunakan dalam mendeskripsikan susunan bilangan dua
dimensi yang disusun dalam baris dan kolom. matriks memiliki ukuran, yaitu jumlah baris dan kolom.

Hubungan antara array dan matriks adalah bahwa matriks adalah bentuk khusus dari array. Array dapat memiliki lebih dari dua dimensi, tetapi matriks selalu memiliki dua dimensi. Dalam pemrograman, array dan matriks sering digunakan untuk menyimpan data dalam jumlah besar dan memudahkan pengaksesan data tersebut.

Mari kita bahas beberapa hal terkait vektor terlebih dahulu

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

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

```
>w=intrandom(10,12)
```

```
[11,  4,  9,  3,  6,  4,  11,  3,  6,  2]
```

Untuk mengurutkan angka acak

```
>sort(v)
```

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

Selanjutnya mengurutkan angka acak dengan menyederhanakan angka yang sama

```
>unique(v)
```

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

Menemukan banyaknya setiap elemen dengan bantuan interval

```
>s=intrandom(10,20)
```

```
[12,  9,  15,  10,  7,  11,  11,  4,  6,  18]
```

```
>x=[5,10,15,20]
```

```
[5,  10,  15,  20]
```

```
>find(x,s)
```

```
[2,  1,  3,  2,  1,  2,  2,  0,  1,  3]
```

Berikutnya adalah cara mencari indeks dari sebuah vektor dengan contoh vekUntuk indeks pada EMT berbeda dengan indeks pada Phyton yang kita pelajari sebelumnya di Algoritma dan pemrograman. Perbedaannya juka sebelumnya untk menentukan indeks akan dimulalai dari nol namun di mmenentukan indeks di EMT akan dimulai dari angka satu, berikut penjelasannya

```
>indexof(w,1:10)
```

```
[0,  10,  4,  2,  0,  5,  0,  0,  3,  0]
```

```
>x= sort(intrandom(10,12))
```

```
[1,  1,  3,  6,  7,  8,  8,  12,  12,  12]
```

```
>indexofsorted(x,1:15)
```

```
[2,  0,  3,  0,  0,  4,  5,  7,  0,  0,  0,  10,  0,  0,  0]
```

```
>z=intrandom(1000,10); multofsorted(sort(z),1:10), sum(%)
```

```
[81,  96,  121,  101,  102,  91,  115,  98,  100,  95]
1000
```

Sampai disini pembahasan terkait dnegan vektor
Selanjutnya kita akan membahas beberapa hal terkait matriks terkait

Untuk Menyimpan Data dalam bentuk Matrik

Pertama, buat sebuah variabel yang akan menampung data matrik, misal X. Variabel ini bebas dengan syarat tidak sama dengan nama fungsi atau konstanta yang sudah ada dalam software.

Selanjutnya,kita akan membuat matrik berordo mxn yang berisi angka

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

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

```
>shortformat; A=random(3,4)
```

```
    0.50136   0.58172   0.02845   0.72032
    0.27668    0.1313   0.84982   0.77608
    0.32956   0.68574   0.42373   0.77217
```

```
>shortformat; A=intrandom(5,4,20)
```

```
          3        18         4        18
          7        19        12         8
          5        17        11        10
         15         1        20         7
         11        13         9         2
```

```
>shortformat; A=redim(1:15,4,4)
```

```
          1         2         3         4
          5         6         7         8
          9        10        11        12
         13        14        15         0
```

```
>(1:5)_2
```

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

```
>random(3,3)_random(2,2)
```

```
    0.74041   0.41322   0.25054
    0.15543   0.10655   0.98859
    0.48475  0.078167   0.57911
    0.46856   0.22056         0
    0.22837   0.47473         0
```

```
>for k=1 to prod(size(A)); A{k}=k; end; short A
```

```
          1         2         3         4
          5         6         7         8
          9        10        11        12
         13        14        15        16
```

```
>B=zeros(size(A))
```

```
          0         0         0         0
          0         0         0         0
          0         0         0         0
          0         0         0         0
```

```
>B=ones(size(A))
```

```
          1         1         1         1
          1         1         1         1
          1         1         1         1
          1         1         1         1
```

Berikutnya operasi penjumlahan dam pengurangan matriks

```
>shortformat; I=intrandom(3,4,10)
```

```
          5         5         7         5
          9         8         3         8
          5         5         1         1
```

```
>shortformat; J=intrandom(3,4,8)
```

```
          7           8           1           3
          3           7           6           7
          1           1           1           6
```

```
>C= I-J
```

```
         -2          -3           6           2
          6           1          -3           1
          4           4           0          -5
```

```
>C= I+J
```

```
         12          13           8           8
         12          15           9          15
          6           6           2           7
```

Dalam materi matriks yang pernah kita pelajari ada sebutan transpose, Invers dan juga determinan, jika menggunakan EMt sebagai berikut secera berurutan:

```
>T = transpose(I)
```

```
          5           9           5
          5           8           5
          7           3           1
          5           8           1
```

```
>T = I'
```

```
          5           9           5
          5           8           5
          7           3           1
          5           8           1
```

```
>K = J^(-1)
```

```
    0.14286       0.125           1     0.33333
    0.33333     0.14286     0.16667     0.14286
          1           1           1     0.16667
```

```
>shortformat; L=intrandom(3,3,7)
```

```
 7          4          4
 3          1          6
 3          7          6
```

```
>det(L)
```

```
-180
```

Selanjutnya adalah cara ekstraksi baris dan kolom, atau sub-matriks,yang mirip dengan R sebagai berikut:

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

```
 4          4
 1          6
 7          6
```

```
>shortformat; X=redim(1:20,4,5)
```

```
  1          2          3          4          5
  6          7          8          9         10
 11         12         13         14         15
 16         17         18         19         20
```

```
>function setmatrixvalue (M, i, j, v) ...
```

```
loop 1 to max(length(i),length(j),length(v))
   M[i{#},j{#}] = v{#};
end;
endfunction
```

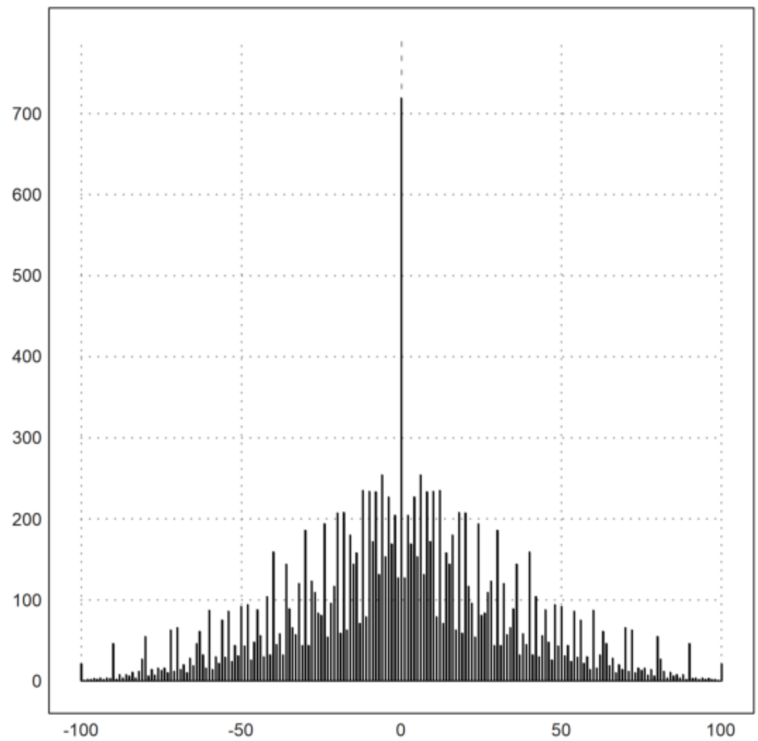
```
>setmatrixvalue(X,1:4,4:-1:1,0); X,
```

```
  1          2          3          0          5
  6          7          0          9         10
 11          0         13         14         15
  0         17         18         19         20
```

```
>(1:4)*(1:4)'
```

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

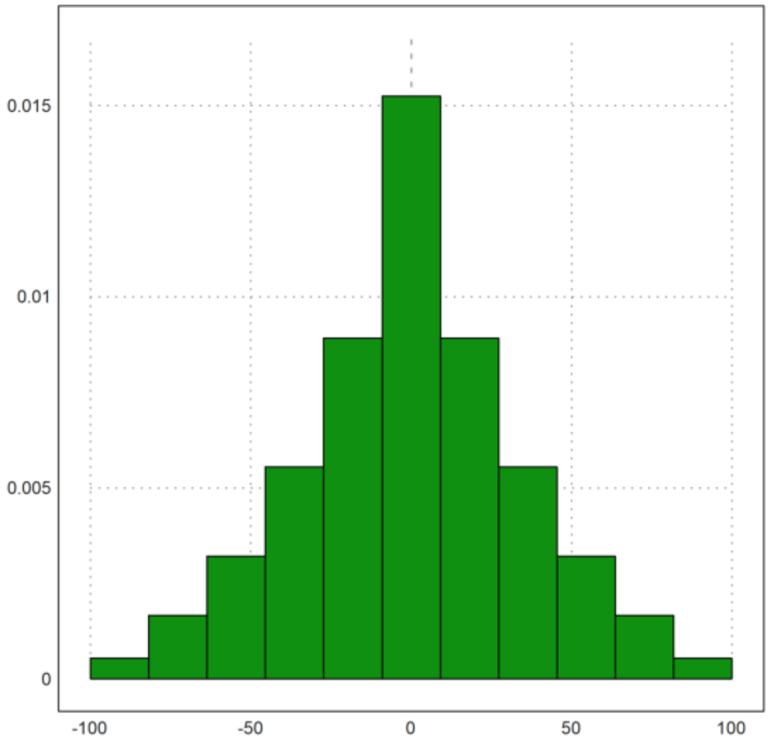
```
>a=0:10; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
>u=sort(unique(q)); f=getmultiplicities(u,q); ...
>statplot(u,f,"h"):
```



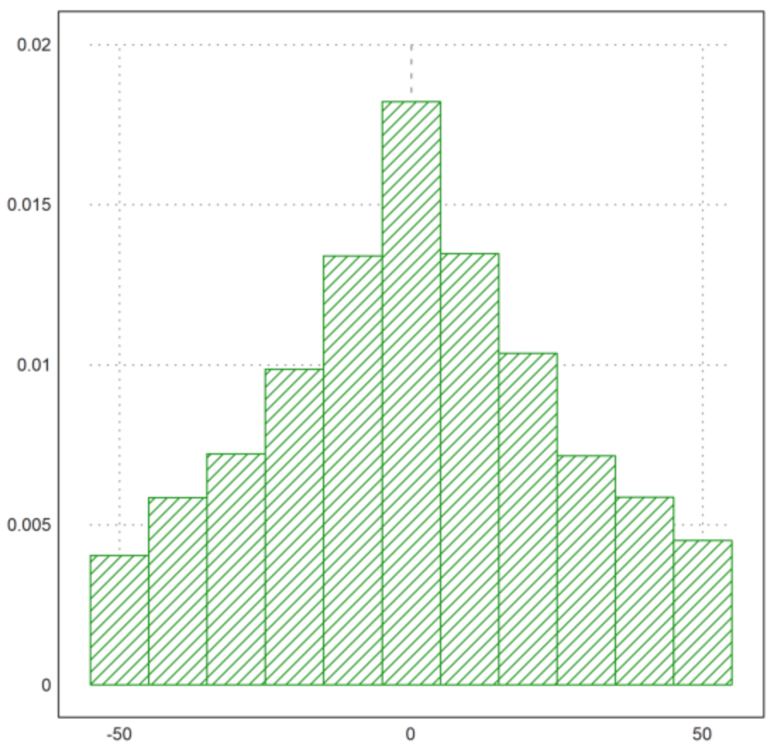

```
>getfrequencies(q,-50:10:50)
```

```
[613,  814,  1088,  1404,  1904,  2389,  1431,  1109,  841,  680]
```

```
>plot2d(q,distribution=11):
```




```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x);
>plot2d(x,y,>bar,style="/"):
```



NAMA : CHINTYA WIJAYANTI
KELAS : MATEMATIKA E
NIM : 22305144029

CAKUPAN MATERI MELIPUTI DIANTARANYA:
-Definisi Bilangan Acak dan Data Acak
-Pengertian Distribusi Diskrit dan Konsep yang Terkait
-Metode Menentukan Distribusi Diskrit

1. Definisi Bilangan Acak dan Data Acak

```
Bilangan Acak adalah bilangan yang tidak dapat diprediksi
```

kemunculannya. Sehingga, tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna.

```
Bilangan acak sendiri dapat dibangkitkan dengan pola tertentu
```

yang dinamakan dengan distribusi, dengan catatan mengikuti fungsi distribusi yang ditentukan.

```
Data acak merupakan hasil dari suatu percobaan acak. Sedangkan
```

percobaan acak adalah suatu proses yang dilakukan sedemikian rupa sehingga hasilnya tidak dapat ditentukan dengan pasti sebelum percobaan tersebut selesai dilakukan

contoh :

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

```
[2,  4,  6,  7,  3,  3,  2,  9,  10,  2]
```

---

2. Pengertian Distribusi Diskrit dan Konsep yang Terkait

```
Distribusi diskrit dalam statistika adalah distribusi data yang
```

memiliki nilai-nilai yang terpisah dan dapat dihitung. Contohnya adalah jumlah anak dalam sebuah keluarga, jumlah mata dadu yang muncul, atau jumlah pelanggan yang datang ke sebuah toko.

```
Distribusi diskrit merujuk pada distribusi
```

probabilitas yang melibatkan variabel acak diskrit. Variabel acak diskrit adalah variabel acak yang hanya dapat mengambil nilai-nilai terpisah, bukan nilai-nilai kontinu seperti pada variabel acak kontinu. Distribusi diskrit memberikan probabilitas masing-masing nilai yang mungkin dari variabel acak tersebut. Berikut adalah beberapa konsep kunci yang terkait dengan distribusi diskrit dalam
statistika:
1. Fungsi Probabilitas Diskrit (Probability Mass Function - PMF):
-Fungsi probabilitas diskrit, atau PMF, memberikan probabilitas bahwa variabel acak diskrit akan mengambil nilai tertentu.
PMF umumnya dilambangkan dengan P(X=x), di mana X adalah variabel acak dan x adalah nilai yang mungkin dari variabel tersebut.

2. Ruang Sampel (Sample Space):
-Ruang sampel adalah himpunan semua hasil mungkin dari suatu percobaan acak yang dapat diukur.
-Setiap elemen dalam ruang sampel merupakan hasil yang mungkin dari variabel acak.

3. Hukum Probabilitas untuk Distribusi Diskrit:
Probabilitas suatu kejadian adalah bilangan yang berada dalam rentang 0 hingga 1, atau 0 <= P(A)<=1 untuk setiap kejadian A.
Probabilitas total dari semua hasil dalam ruang sampel adalah 1, atau P (S)= 1, di mana S adalah ruang sampel.

4. Fungsi Distribusi Kumulatif (Cumulative Distribution Function - CDF):
-Fungsi distribusi kumulatif memberikan probabilitas bahwa variabel acak diskrit kurang dari atau sama dengan nilai tertentu.
-Notasi matematisnya sering kali disimbolkan sebagai F(x)-P(X<=x)

5. Harapan (Expectation) dan Varians:
-Harapan atau nilai rata-rata (E(X)) dari distribusi diskrit adalah
jumlah tertimbang dari nilai-nilai mungkin berdasarkan probabilitas
masing-masing nilai.
-Varians Var(X)) mengukur sejauh mana nilai-nilai distribusi tersebar
dari nilai rata-ratanya.

```
>
```

---

3. Metode Menentukan Distribusi Diskrit

```
Untuk menentukan distribusi diskrit sendiri, dapat menggunakan
```

metode berikut. Pertama kita mengatur fungsi distribusi, fungsi distribusi adalah fungsi yang menggambarkan kemungkinan suatu variabel acak untuk memiliki nilai tertentu atau dalam rentang waktu tertentu.
Langkah mengatur fungsi distribusi:
-Menentukan jenis var acak yg akan diteliti, apakah diskrit atau
kontinu
-Menentukan parameter-parameter yang berkaitan dengan fungsi
distribusi, spt probabilitas
-Menentukan bentuk fungsi distribusi yg sesuai dg variabel acak dan
parameter yg sudah ditentukan

```
>wd = 0|((1:6)+[-0.01,0.01,0,0,0,0])/5
```

```
[0,  0.198,  0.402,  0.6,  0.8,  1,  1.2]
```

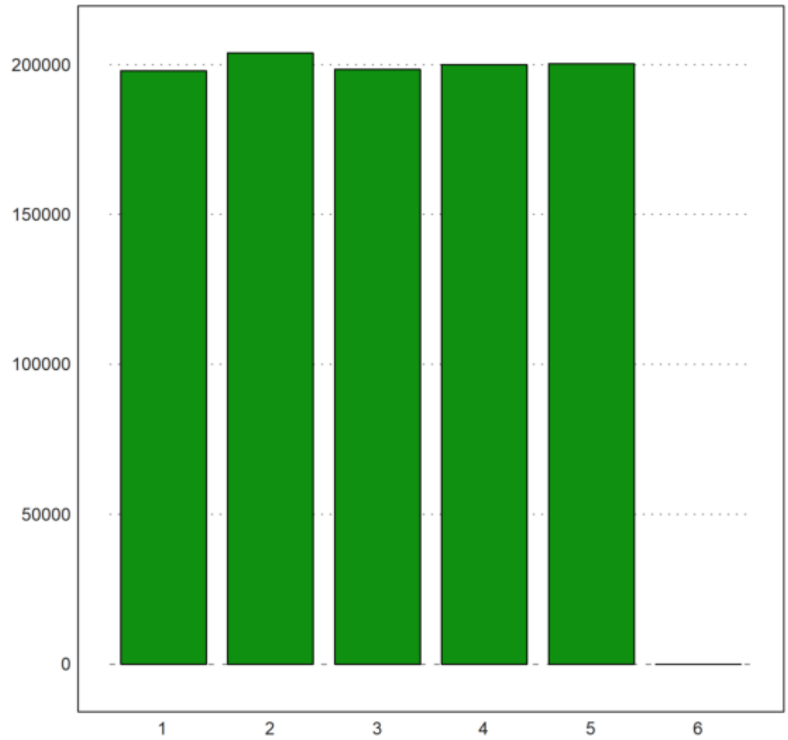
Artinya dengan probabilitas wd[i+1]-wd[i] kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator angka acak untuk ini. Fungsi find(v,x) menemukan nilai x dalam vektor v. Fungsi ini juga berlaku untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya sangat halus sehingga melihatnya hanya dengan iterasi yang sangat banyak.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai 1...K dalam v. menerima hasilnya, jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1)
```

```
K=max(v); n=cols(v);
fr=getfrequencies(v,1:K);
return max(fr/n-1/K)<delta/sqrt(n);
endfunction
```

Memang fungsi menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

```
0
```

Dan itu menerima generator acak bawaan.

```
>checkrandom(intrandom(1,1000000,6))
```

```
1
```

Kita dapat menghitung distribusi binomial. Pertama ada binomialsum(), yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

```
>bindis(410,1000,0.4)
```

```
0.7514
```

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Level default adalah alfa.

Arti interval ini adalah jika p berada di luar interval, hasil pengamatan 410 dalam 1000 jarang terjadi.

```
>clopperpearson(410,1000)
```

```
[0.37932,  0.44121]
```

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas.
Tapi untuk n besar, penjumlahan langsungnya tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
0.7514
```

invbinsum() menghitung kebalikan dari binomialsum().

```
>2*hypergeomsum(1,5,13,26)
```

```
0.32174
```

Ada juga simulasi distribusi multinomial.Distribusi diskrit dalam statistika adalah distribusi data yang memiliki nilai-nilai yang terpisah dan dapat dihitung. Contohnya adalah jumlah anak dalam sebuah keluarga, jumlah mata dadu yang muncul, atau jumlah pelanggan yang datang ke sebuah toko.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

```
        385          95         520
        407          88         505
        388         106         506
        381         114         505
        413         113         474
        388          99         513
        396          91         513
        399          93         508
        390          89         521
        402         103         495
```

Contoh Soal

Simulasikan 1000 data acak dengan distribusi normal dengan mean 1 dan simpangan baku 2. Hitung rata-rata!

Jawab :
// Simulasi data acak dengan distribusi normal

```
>data = randnormal(1,1000,2)
```

```
 [3.9181,  1.9527,  -0.33732,  2.2526,  1.221,  1.5064,  1.0475,
 2.1997,  2.5056,  1.9128,  1.727,  2.1483,  2.2224,  3.6901,  1.1749,
 1.9995,  0.24513,  2.3104,  3.1337,  0.38286,  2.8783,  1.2136,
 2.0146,  -0.097474,  2.1818,  0.0020967,  3.6067,  2.2221,  1.6357,
 0.73029,  3.1022,  2.1763,  2.6671,  2.3198,  2.326,  1.5059,  1.4371,
 2.7432,  0.1283,  1.6684,  1.9932,  0.76937,  2.134,  1.7466,  1.3792,
 1.5807,  0.83863,  2.5515,  3.2569,  4.4061,  1.9229,  3.8785,
 1.5262,  2.3406,  1.8594,  2.4003,  2.8752,  2.5498,  2.2527,  2.7602,
 2.6761,  1.6431,  3.4518,  2.2219,  1.7896,  2.519,  2.2191,  2.1538,
 2.4901,  1.8535,  3.1297,  1.7501,  4.6754,  3.3252,  1.7295,  1.4201,
 2.515,  2.5155,  0.51684,  1.727,  1.4341,  3.1193,  1.0781,  1.1891,
 0.44626,  3.7094,  1.1403,  0.064038,  0.497,  2.4713,  2.0817,
 2.2472,  4.3022,  3.5434,  2.9278,  2.5287,  1.1453,  1.9166,  2.1864,
 1.8073,  2.3455,  0.32874,  2.8625,  1.8259,  1.6132,  1.9987,
 2.8166,  4.1364,  1.2717,  3.2232,  1.1259,  1.2758,  1.5482,  3.7335,
 1.509,  3.0431,  0.60999,  1.0187,  1.8762,  1.9697,  1.6101,  1.8664,
 2.1309,  2.2315,  2.0468,  1.559,  3.853,  1.89,  1.5104,  2.1613,
 1.5506,  5.3184,  2.3794,  2.3148,  2.5744,  1.7659,  2.6473,
 0.66693,  2.6216,  4.014,  3.6484,  -0.083378,  3.5684,  2.7952,
 2.1936,  2.381,  1.8078,  0.78504,  3.4552,  1.2495,  0.88303,
  ... ]
```

```
>mean(data)
```

```
1.9987
```

Nama: Bintang Mahija Aryacetta
NIM : 22305144003
Kelas : Matematika E 2022

Membaca Data yang Tersimpan di dalam Berkas dengan Berbagai Format(teks biasa, CSV) untuk di analisis lebih lanjut

membaca data dari teks biasa

pertama-tama, kita akan mencoba membaca data dengan teks biasa yang terdapat pada contoh yang telah di berikan pada besmart yaitu dengan menuliskan fungsi ”printfile(nama file teks biasa, berapa baris yang akan di print)”

seperti di bawah kita akan print tabel pada data yang terdapat pada buku online ”Einführung in die Statistik mit R” oleh A. Handl.

```
>printfile("table.dat",4)
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

pada kasus ini saya telah menunjukan terdapat 4 baris yang telah di print pada emt yang berisi angka dan token(string)

kali ini kita akan membaca tabel dengan lebih mudah atau dengan bahasa kita sendiri. Untuk ini, kita mendefinisikan set token. Fungsi strtokens () mendapatkan vektor string token dari string tertentu.

```
>mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
```

sekarang kita dapat membacanya dengan cara kita sendiri

Argumen tok2, tok4, dll. Adalah definisi dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable (), jadi Anda perlu memberinya ”: =” untuk mendefinisikannya.

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
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;
```

lalu kita akan print tabel sesuai dengan tabel awal namun dengan bentuk tabel yang berbeda

```
>writetable(MT[1:6],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
MT is not a variable!
Error in:
writetable(MT[1:6],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok ...
                  ^
```

Titik "." mewakili nilai-nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan terlebih dahulu, kita hanya perlu menentukan, kolom mana yang berisi token dan bukan angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Fungsi readtable () sekarang mengembalikan satu set token.

```
>tok
```

```
Variable tok not found!
Error in:
tok ...
   ^
```

Tabel berisi entri dari file dengan token yang diterjemahkan menjadi angka.

String khusus NA = "." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
>MT[1]
```

```
MT is not a variable!
Error in:
MT[1] ...
     ^
```

Berikut adalah isi tabel dengan bilangan yang belum diterjemahkan.

```
>writetable(MT[1:6],wc=5)
```

```
MT is not a variable!
Error in:
writetable(MT[1:6],wc=5) ...
                  ^
```

Untuk kenyamanan, Anda bisa 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 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); ...
                ^
```

tabel sudah dapat di analisis lebih lanjut

```
>
```

pertama tama kita download file csv yang telah di sediakan di besmart, setelah itu kita jadi satukan dalam 1 folder dengan file emt kita. lalu masukan file tersebut dengan definisi file="nama file csv"

```
>file="test.csv";  ...
>M=random(3,3); writematrix(M,file)
```

M mendefinisikan sebagai matrix
random(n,m) mendefinisikan matrix dengan variabel acak yang akan di keluarakan
writematrix digunakan untuk menuliskan matriks yang ada

lalu kita print datanya dengan

```
>printfile(file)
```

```
0.4191983703672241,0.4504034185243261,0.8530870403686531
0.7175631662797505,0.6559204322999515,0.883095901954524
0.07277738544441656,0.8168773103666358,0.5703787256414963
```

titik desimal pada data tersebut dapat di jadikan pada format EMT dengan cara menggunakan readmatrix()

```
>readmatrix(file)
```

```
     0.4192      0.4504     0.85309
    0.71756     0.65592      0.8831
   0.072777     0.81688     0.57038
```

```
>
```

Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai dipisahkan koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut ini contohnya.

```
>printfile("excel-data.csv")
```

```
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 perlu membaca matriks ke EMT.

Cara termudah untuk membaca ini ke dalam Euler adalah readmatrix (). Semua koma diganti dengan titik dengan parameter> koma. Untuk CSV bahasa Inggris, cukup abaikan parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

```
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;
```

data siap di analisis lebih lanjut

```
>reset;
```

nomer 1

```
>file="sample.csv"
```

```
sample.csv
```

```
>printfile(file,7)
```

```
Could not open the file
sample.csv
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>ctok=[1]; {MT,hd,tok}=readtable("sample.csv",ctok=ctok);
```

```
Could not open the file
sample.csv
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

```
>tok
```

```
Variable tok not found!
Error in:
tok ...
   ^
```

```
>MT[2]
```

```
MT is not a variable!
Error in:
MT[2] ...
     ^
```

```
>writetable(MT[1:6],wc=5)
```

```
MT is not a variable!
Error in:
writetable(MT[1:6],wc=5) ...
                  ^
```

```
>Table={{readtable(file,ctok=ctok)}};
```

```
Could not open the file
sample.csv
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

```
>writetable(Table,ctok=ctok,wc=10);
```

```
Variable or function Table not found.
Error in:
writetable(Table,ctok=ctok,wc=10); ...
                ^
```

```
>reset;
```

---

nomer 2

```
>file="test.dat"
```

```
test.dat
```

```
>printfile(file,7)
```

```
Could not open the file
test.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>mf:=["m","f"];
>{MT,hd}=readtable("table1.dat",tok2:=mf);
```

```
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;
```

```
>load over statistics;
>writetable(MT[1:6],labc=hd,wc=5,tok2:=mf);
```

```
MT is not a variable!
Error in:
writetable(MT[1:6],labc=hd,wc=5,tok2:=mf); ...
                  ^
```

```
>reset;
```

BAB VISUALISASI DAN KOMPUTASI STATISTIKA DENGAN EMT

Nama : Adib Brian Syuhada
Kelas : Matematika E 2022
NIM : 22305144014

Pada sesi ini akan membahas mengenai 2 subbab yang merupakan lanjutan dari subbab sebelumnya, yaitu :
1. Membaca data dari internet
2. Perhitungan terkait analisis data statistika deskriptif meliputi rata-rata, simpangan baku, jangkauan, modus, ukuran data,varians dan median.
Membaca data dari internet

Situs web atau file dari URL dapat dibuka dengan menggunakan EMT dan dapat dibaca baris demi baris.

Berikut contoh penggunaan EMT untuk membuka url untuk mengetahui versi dari EMT

```
>function readversion () ...
```

```
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html"); //membuka url
repeat //loop yang berlangsung sampai akhir file url
until urleof();
s=urlgetline();  //membaca baris teks
k=strfind(s,"Version",1);  //mencari substring"Version". jika ditemukan akan disimpan di k
if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif; //berhenti sebelum <
end;
urlclose();
endfunction
```

```
> readversion
```

```
Version 2022-05-18
```

```
>function readdataurl () ...
```

```
urlopen("https://kumparan.com/berita-terkini/3-contoh-soal-desil-data-tunggal-beserta-kunci-jawabanny
repeat
until urleof();
s=urlgetline();
k=strfind(s,"Tentukan persentil",1);
if k>0 then substring(s,k,strfind(s,".",k)-1), break; endif;
end;
urlclose();
endfunction
```

Selanjutnya kita mencoba dengan cara yang sama untuk mengambil soal dari website yang ada di internet

```
>readdataurl
```

Analisis data statistika deskriptif

Statistika deskriptif adalah bidang ilmu statistika yang mempelajari cara-cara untuk pengumpulan, penyusunan, dan penyajian data sehingga memberikan informasi yang berguna. Perlu diketahui juga bahwa statistika deskriptif memberikan informasi hanya mengenai data yang dipunyai dan sama sekali tidak menarik inferensia atau kesimpulan apapun tentang gugus data induknya yang lebih besar.

Dalam praktiknya,analisis data statistika deskriptif bisa dilakukan dengan menerapkan sejumlah metode statistik, seperti :

1. Mencari rata rata/mean

Metode pertama yang digunakan untuk melakukan analisis statistika adalah mean atau sering disebut rata-rata. Saat akan menghitung rata-rata, kita bisa melakukan dengan cara menambahkan daftar angka kemudian membagi angka tersebut dengan jumlah item dalam daftar. Metode ini memungkinkan penentuan tren keseluruhan dari kumpulan data dan mampu mendapatkan tampilan data yang cepat dan ringkas. Manfaat dari metode ini juga termasuk perhitungan yang sederhana dan cepat.

a. Rata-rata hitung data tunggal

Misalkan

$$x_1, x_2, x_3, ..., x_n$$

adalah data yang dikumpulkan dari suatu sampel atau populasi maka rata-rata hitung untuk sampel disimbolkan dengan

$$\bar{x}$$

dan rata-rata hitung untuk populasi disimbolkan dengan

$$\mu$$

Sehingga, untuk mencari rata-rata hitung data tunggal terdapat 2 jenis rumus sebagai berikut :
1. Rata-rata hitung sampel

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

2. Rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^{n} X_i}{n}$$

Untuk menghitung rata-rata data tunggal dengan EMT, kita dapat menggunakan sintaks

> mean ([data])

Contoh Soal:
1. Diketahui data usia(dalam tahun) penduduk suatu daerah adalah sebagai berikut:
60,70,66,75,77,68,45,30,15,71,69,84,13
hitunglah rata-rata usia penduduk tersebut.
Jawab :

```
>mean([60,70,66,75,77,68,45,30,15,71,69,84,13])
```

```
57.1538461538
```

Jadi, rata rata data tersebut adalah 57.1538461538

2. Nilai ulangan matematika dari 10 siswa adalah 80, 88, 70, 60, 90, 75, 92, 78, 67, 90. Tentukan rata-rata dari data tersebut!
Jawab :

```
>mean([80, 88, 70, 60, 90, 75, 92, 78, 67, 90])
```

```
79
```

Jadi, rata-rata dari data tersebut yaitu 79

b. Rata-rata data tabel distribusi

Jika diberikan data

$$x_1, x_2, ..., x_n$$

yang memiliki frekuensi berturut- turut

$$f_1, f_2, ..., f_n$$

maka, rataan hitung dari data yang disajikan dalam daftar distribusi tersebut ditentukan dengan 2 jenis rumus sebagai berikut :

1. Rata-rata hitung sampel
Untuk rata-rata hitung sampel,

$$\bar{x} = \frac{\sum_{i=1}^{n} f_i x_i}{\sum_{i=1}^{n} f_i}$$

2.Rata-rata hitung populasi
Untuk rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^{n} f_i x_i}{\sum_{i=1}^{n} f_i}$$

Cara diatas adalah beberapa perhitungan untuk mencari rata-rata data tabel distribusi menggunakan metode yang ada dalam statistika. Dengan menggunakan EMT kita juga bisa menghitung rata-rata data tabel distribusi dengan mudah, yaitu dengan cara berikut:
1. Mendeskripsikan data dan frekuensi
2. Menghitung rata-rata menggunakan perintah berikut :

> mean(data,frekuensi)

Contoh soal:
Diberikan data berat badan siswa kelas V SD yang memiliki jumlah siswa sebanyak 35 orang anak. anak dengan berat 30kg terdapat 5 orang, anak dengan berat 35kg terdapat 11 orang, anak dengan berat 40kg terdapat 4 orang, anak dengan berat 38kg terdapat 7 orang, anak dengan berat 44kg terdapat 7 orang, dan anak dengan berat 50kg terdapat 1 orang. Tentukan rata-rata berat siswa kelas V SD tersebut!
Jawab :

```
>printfile("tabel berat badan kelas V SD.dat",7); //meringkas informasi pada soal dengan membuat tabel
```

```
Could not open the file
tabel berat badan kelas V SD.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>data=[30,35,38,40,44,50]//mendefinisikan data sebagai berat siswa dalam satuan kilogram
```

```
[30,  35,  38,  40,  44,  50]
```

```
>frekuensi=[5,11,7,4,7,1]//mendefinisikan frekuensi sebagai banyak siswa
```

```
[5,  11,  7,  4,  7,  1]
```

```
>mean(data,frekuensi) //menghitung rata-rata
```

```
37.6857142857
```

Jadi, rata-rata berat badan siswa SD kelas V adalah 37.6857142857

c. Rata-rata hitung data kelompok

Misalkan suatu data kelompok terdiri dari n kelas dengan nilai tengah masing-masing kelas secara berturut-turut adalah

$$t_1, t_2, ..., t_n$$

dan masing-masing frekuensinya adalah

$$f_1, f_2, ..., f_n$$

Untuk mencari rata rata hitung data tersebut terdapat 2 jenis rumus sebagai berikut :
1. Rata-rata hitung sampel
untuk rata-rata hitung sampel,

$$\bar{x} = \frac{\sum_{i=1}^{n} t_i f_i}{\sum_{i=1}^{n} f_i}$$

2. Rata-rata hitung populasi
untuk rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^{n} t_i f_i}{\sum_{i=1}^{n} f_i}$$

Untuk menghitung rata-rata data kelompok di EMT dapat dilakukan dengan langkah berikut :
1. Menentukan tepi bawah kelas(Tb), panjang kelas(P), dan tepi atas kelas(Ta) dengan rumus :

$$Tb = a - 0,5$$

$$P = (b - a) + 1$$

$$Ta = b + 0,5$$

Keterangan :
a = batas bawah kelas
b = batas atas kelas

2. Membuat data menjadi bentuk tabel, dengan perintah

> r= tepi bawah terkecil : panjang kelas : tepi atas terbesar;
f=[frekuensi];
>T:r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' |f';
writetable(T, labc=["tepi bawah", "tapi atas", "frekuensi"])

3. Menghitung nilai tengah kelas, dengan perintah

>T[,1]+T[,2]/2

4. Mengubah baris menjadi kolom

>t=fold(r,[0.5,0.5])

5. Menghitung rata-rata, dengan perintah

>mean(t,f)

Contoh soal :
1. Disajikan data kelompok seperti berikut :

```
>printfile("Tabel rata-rata data kelompok.dat",7)
```

```
Could not open the file
Tabel rata-rata data kelompok.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>31-0.5  //Tepi bawah terkecil
```

```
30.5
```

```
>(40-31)+1  //Panjang kelas
```

```
10
```

```
>90+0.5 //Tepi atas kelas
```

```
90.5
```

```
>r=30.5:10:90.5; f=[3, 5, 10, 11, 8, 3];
>T:=r[1:6]' | r[2:7]' | f' ; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

```
  tepi bawah tepi atas frekuensi
        30.5      40.5         3
        40.5      50.5         5
        50.5      60.5        10
        60.5      70.5        11
        70.5      80.5         8
        80.5      90.5         3
```

```
>t=(T[,1]+T[,2])/2  //menghitung nilai tengah kelas
```

```
          35.5
          45.5
          55.5
          65.5
          75.5
          85.5
```

```
>t=fold(r,[0.5,0.5]) // mengubah tampilan data kolom menjadi baris dan sebaliknya
```

```
[35.5,  45.5,  55.5,  65.5,  75.5,  85.5]
```

```
>mean(t,f)
```

```
61.75
```

Jadi, rata-rata data kelompok tersebut adalah 61,75

2. Diberikan data kelompok berikut yang mewakili jumlah jam belajar per minggu dari sekelompok siswa :

```
>printfile("Tabel data kelompok conso 2.dat",5)
```

```
Could not open the file
Tabel data kelompok conso 2.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

Hitunglah rata-rata jumlah jam belajar per minggu dari data kelompok tersebut!
Jawab :

```
>10-0.5  //tepi bawah terkecil
```

```
9.5
```

```
>(14-10)+1  //panjang kelas
```

```
5
```

```
>29+0.5  //tepi atas terbesar
```

```
29.5
```

```
>r=9.5:5:29.5; f=[5, 8, 12, 6];
>T:=r[1:4]' | r[2:5]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

```
 tepi bawah tepi atas frekuensi
        9.5      14.5         5
       14.5      19.5         8
       19.5      24.5        12
       24.5      29.5         6
```

```
>t=(T[,1]+T[,2])/2  // menghitung nilai tengah kelas
```

```
12
17
22
27
```

```
>t=fold(r,[0.5,0.5]) // mengubah tampilan data kolom menjadi baris dan sebaliknya
```

```
[12,  17,  22,  27]
```

```
>mean(t,f)
```

```
20.064516129
```

Jadi, rata-rata data kelompok tersebut yaitu 20.064516129

2. Mencari median

Median (Me) adalah nilai tengah dari suatu data yang telah disusun dari data terkecil sampai data terbesar atau sebaliknya. Selain sebagai ukuran pemusatan data, median juga dijadikan sebagai ukuran letak data dan dikenal sebagai kuartil 2 (Q2). Rumus perhitungan median dibedakan untuk data tak berkelompok dan data berkelompok.

a. Median data tunggal

Median data tunggal adalah mengurutkan data berdasarkan nilainya, misalkan data yang telah terurut dari data terkecil ke data terbesar adalah

$$x_1, x_2, ..., x_n$$

`untuk menentukan letak median dengan menggunakan rumus :`

1. Jika jumlah suatu data(n) berjumlah ganjil maka nilai mediannya adalah sama dengan data yang memiliki nilai di urutan paling tengah yang memiliki nomor urut k, dimana untuk menentukan nilai k dapat dihitung menggunakan rumus:

$$k = \frac{n+1}{2}$$

2. Jika jumlah suatu data (n) berjumlah genap, maka untuk menghitung mediannya dengan menggunakan rumus :

$$k = \frac{n}{2}$$
$$Median = \frac{1}{2}(x_k + x_{k+1})$$

Diatas adalah rumus untuk mencari median secara statistika. Dengan menggunakan EMT kita bisa menentukan median dengan menggunakan perintah

> median([data])

perintah tersebut dapat berjalan dengan baik apabila data sudah diurutkan terlebih dahulu dari data terkecil hingga terbesar.
Contoh soal :
Diketahui data hasil tes SKD calon PNS adalah sebagai berikut :
487, 300, 450, 500, 521, 440
Tentukan nilai median dari data tersebut!
Jawab :

```
>data=[487, 300, 450, 500, 521, 440]; //mendeskripsikan data
>urutan=sort(data)  //mengurutkan data
```

```
[300,  440,  450,  487,  500,  521]
```

```
>median([urutan])
```

```
468.5
```

Jadi, nilai median dari data hasil tes SKD adalah 468.5

b. Median data kelompok

Menghitung median data kelompok dapat menggunakan rumus di bawah ini :

$$M_e = Tb + p\frac{\frac{1}{2}n - F}{f}$$

Keterangan:
Tb = tepi bawah kelas median, ialah kelas dimana median terletak
p = panjang kelas median
n = ukuran sampel / banyak data
F = jumlah semua frekuensi dengan tanda kelas lebih kecil dari tanda kelas median.
f = frekuensi kelas median

Untuk menghitung median data berkelompok di EMT, dapat dilakukan dengan cara berikut:
1. Menentukan tepi bawah kelas (Tb), panjang kelas (P), dan tepi atas kelas (Ta) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"]))

3. Mendeskripsikan batas bawah kelas median, panjang kelas median, banyak data, jumlah frekuensi sebelum kelas median, frekuensi median

> Tb=(tepi bawah kelas median), p=(panjang kelas median), n=(banyak data), F=(jumlah frekuensi sebelum kelas median), f=(frekuensi kelas median)

4. Menghitung median data dengan perintah:

> Tb+p*(1/2*n-F)/f

Contoh soal :
Berikut adalah data hasil dari pengukuran berat badan 20 siswa SD kelas V. Dari ke 20 siswa, siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 4 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 3 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 2 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 3 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 3 orang. Tentukan median dari
data hasil pengukuran berat badan 20 siswa di SD tersebut!
Penyelesaian:
Menentukan tepi bawah kelas yang terkecil

```
>21-0.5 // menentukan tepi bawah kelas terkecil
```

```
20.5
```

```
> (26-21)+1 // menentukan panjang kelas
```

```
6
```

```
> 56+0.5 // tepi atas kelas terbesar
```

```
56.5
```

```
>r=20.5:6:56.5; f=[5, 4, 3, 2, 3, 3];
>T :=r[1:6]' | r[2:7]' | f' ; writetable(T, labc=["Tb", "Ta", "frekuensi"])
```

```
   Tb        Ta frekuensi
 20.5      26.5         5
 26.5      32.5         4
 32.5      38.5         3
 38.5      44.5         2
 44.5      50.5         3
 50.5      56.5         3
```

```
>Tb=32.5, p=6, n=20, F=9, f=3
```

```
32.5
6
20
9
3
```

```
>Tb+p*(1/2*n-F)/f
```

```
34.5
```

Jadi, median dari data hasil pengukuran berat badan 20 siswa SD kelas V adalah 34.5

3. Mencari Modus

Modus adalah area fokus dalam analisis statistika deskriptif yang termasuk dalam ukuran pusat data. Ini adalah nilai yang paling sering muncul dalam kumpulan data atau nilai yang memiliki frekuensi tertinggi dalam distribusi data. Modus dapat dibagi menjadi dua jenis, yaitu modus untuk data tunggal dan modus untuk data kelompok.

a.Modus untuk data tunggal:

Menentukan modus untuk data tunggal cukup sederhana. Pertama, data diurutkan dari nilai terkecil ke terbesar sehingga data dengan nilai yang sama berdekatan satu sama lain. Selanjutnya, frekuensi masing-masing data dihitung, dan data yang memiliki frekuensi tertinggi dipilih sebagai modus.

b.Modus untuk data kelompok

Berikut rumus untuk mencari modus data kelompok :

$$M_o = Tb + \frac{d_1}{d_1 + d_2} c$$

Keterangan :
Tb = Tepi bawah
d1 = selisih f modus dengan f sebelumnya
d2 = selisih f modus dengan f sesudahnya

```
c = Panjang kelas
```

Untuk menghitung modus data berkelompok di EMT, dapat dilakukan dengan cara berikut:
1. Menentukan tepi bawah kelas (Tb), panjang kelas (P), dan tepi atas kelas (Ta) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dimana a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; v=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"]))

3. Mendeskripsikan tepi bawah kelas modus, panjang kelas modus,selisih frekuensi modus dengan frekuensi sebelumnya, selisih frekuensi modus dengan frekuensi sesudahnya
> Tb=(tepi bawah kelas modus), p=(panjang kelas modus), d1=(selisih frekuensi modus dengan frekuensi sebelumnya), d2=(selisih frekuensi dengan frekuensi sesudahnya)

4. Menghitung modus dengan perintah:
> Tb+p*d1/(d1+d2)

Contoh soal
Diketahui sebuah data kelompok sebagai berikut :

```
>printfile("Tabel modus data kelompok.dat",8)
```

```
Could not open the file
Tabel modus data kelompok.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

Berapakah modus dari data tersebut?

```
>20-0.5  //menentukan tepi bawah kelas
```

```
19.5
```

```
>(29-20)+1 //menentukan panjang kelas
```

```
10
```

```
>89+0.5 //menentukan tepi atas
```

```
89.5
```

```
>r=19.5:10:89.5; f=[3, 7, 8, 12, 9, 6, 5];
>T:=r[1:7]' | r[2:8]' | f'; writetable(T,labc=["Tb", "Ta", "frekuensi"])
```

```
        Tb        Ta frekuensi
      19.5      29.5         3
      29.5      39.5         7
      39.5      49.5         8
      49.5      59.5        12
      59.5      69.5         9
      69.5      79.5         6
      79.5      89.5         5
```

Berdasarkan tabel di atas, modus berada pada kelas 49.5-59.5

```
>Tb=49.5, p=10, d1=12-8, d2=12-9
```

```
49.5
10
4
3
```

```
>Tb+p*d1/(d1+d2)
```

```
55.2142857143
```

Jadi, modus dari data kelompok di atas adalah 55.2142857143

4. Mencari varians/ragam

Varians digunakan untuk mengetahui bagaimana sebaran data terhadap mean atau nilai rata-rata. Sederhananya, varians adalah ukuran statistik jauh dekatnya penyebaran data dari nilai rata-ratanya. Dalam mencari ragam dapat dikelompokkan menjadi 2 jenis yaitu sebagai berikut :

Rumus untuk varians data tunggal berikut :
1) Untuk populasi

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x-\mu)^2}{n}$$

2) Untuk sampel

$$S^2 = \frac{\sum_{i=1}^{n}(x-\bar{x})^2}{n-1}$$

Pada EMT, untuk menemukan suatu Ragam data tunggal dapat menggunakan perintah berikut:

> mean(dev^2)

Contoh soal
Hitunglah nilai varians dari data sampel nilai siswa: 9, 10, 6, 7!

```
>data=[9, 10, 6, 7]; //mendefinisikan data
>urut=sort(data) //mengurutkan data
```

```
[6,  7,  9,  10]
```

```
>xbar=mean(urut) //menghitung rata rata dari data
```

```
8
```

```
>dev= urut-xbar
```

```
[-2,  -1,  1,  2]
```

```
>varians=mean(dev^2) //menghitung varians
```

```
2.5
```

Jadi, varians dari data sampel tersebut adalah 2.5

Rumus untuk varians data kelompok sebagai berikut :
1) Untuk populasi

$$\sigma^2 = \frac{\sum_{i=1}^{n} f_i (x_i - \mu)^2}{\sum_{i=1}^{n} f_i}$$

2) Untuk sampel

$$S^2 = \frac{\sum_{i=1}^{n} f_i (x_i - \bar{x})^2}{\sum_{i=1}^{n} f_i - 1}$$

Pada EMT, untuk menemukan Ragam data berkelompk dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas (Tb), panjang kelas (P), dan tepi atas kelas (Ta) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])

3. Menghitung Ragam dengan perintah

> (T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]);m=mean(t,f);
> sum(f*(t-m)^2)/sum(f) //untuk populasi
> sum(f*(t-m)^2)/(sum(f)-1) //untuk sampel

Contoh soal
Tentukan varians data sampel dari tabel berikut :

```
>printfile("Tabel data kelompok varians.dat",7)
```

```
Could not open the file
Tabel data kelompok varians.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>63-0.5  //tapi bawah terkecil
```

```
62.5
```

```
>(67-63)+1  //panjang kelas
```

```
5
```

```
>92+0.5  //tepi atas terbesar
```

```
92.5
```

```
>r=62.5:5:92.5; f=[3,2,7,3,4,1];
>T:=r[1:6]' | r[2:7]' | f'; writetable(T, labc=["tepi bawah", "tepi atas", "frekuensi"])
```

```
 tepi bawah tepi atas frekuensi
       62.5      67.5         3
       67.5      72.5         2
       72.5      77.5         7
       77.5      82.5         3
       82.5      87.5         4
       87.5      92.5         1
```

```
>(T[,1]+T[,2])/2; t=fold(r,[0.5,0.5])
```

```
[65,  70,  75,  80,  85,  90]
```

```
>m=mean(t,f)
```

```
76.5
```

```
>sum(f*(t-m)^2)/(sum(f)-1)
```

```
52.8947368421
```

Jadi, varians dari data kelompok dari tabel di atas adalah 52.8947368421

```
>
```

5. Mencari Simpangan Baku

Standar Deviasi atau simpangan baku adalah akar dari ragam/varians. Untuk nenetukan nilai standar deviasi, caranya:

$$\sigma = \sqrt{\sigma^2}$$

$$atau$$

$$S = \sqrt{S^2}$$

a. Simpangan baku data tunggal

Untuk data tunggal, simpangan baku populasi atau sampel dapat dirumuskan sebagai berikut:

1) Untuk populasi

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x-\mu)^2}{n}}$$

2) Untuk sampel

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x-\bar{x})^2}{n-1}}$$

Pada EMT, untuk menemukan suatu Ragam data tunggal dapat menggunakan perintah berikut:

> sqrt(mean(dev^2))

Contoh soal :
1. Simpangan baku untuk data 70,80,40,25,65,87,97,59,24,77,45 adalah
Jawab :

```
>data=[70,80,40,25,65,87,97,59,24,77,45];
>urut=sort(data)
```

```
[24,  25,  40,  45,  59,  65,  70,  77,  80,  87,  97]
```

```
>x=mean(urut)
```

```
60.8181818182
```

```
>dev=urut-x
```

```
[-36.8182,  -35.8182,  -20.8182,  -15.8182,  -1.81818,  4.18182,
9.18182,  16.1818,  19.1818,  26.1818,  36.1818]
```

```
>varians=mean(dev^2)
```

```
550.148760331
```

```
>simpanganbaku= sqrt(varians)
```

```
23.4552501656
```

Jadi, simpang baku data tersebut adalah 23.4552501656

b. Simpangan baku data kelompok

```
Untuk data berkelompok dapat dirumuskan seperti berikut:
```

1) Untuk populasi

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} f_i(x_i - \mu)^2}{\sum_{i=1}^{n} f_i}}$$

2) Untuk sampel

$$S = \sqrt{\frac{\sum_{i=1}^{n} f_i(x_i - \bar{x})^2}{\sum_{i=1}^{n} f_i - 1}}$$

Pada EMT, untuk menemukan Ragam data berkelompk dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas (Tb), panjang kelas (P), dan tepi atas kelas (Ta) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])

3. Menghitung Ragam dengan perintah

> (T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]);m=mean(t,f);
> sqrt(sum(f*(t-m)^2)/sum(f)) // untuk populasi
> sqrt(sum(f*(t-m)^2)/(sum(f)-1)) // untuk sampel

Contoh soal :
Simpangan baku dari tabel dibawah ini adalah

```
>printfile("Tabel simpangan baku data kelompok.dat",7)
```

```
Could not open the file
Tabel simpangan baku data kelompok.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>41-0.5 //tepi bawah terkecil
```

```
40.5
```

```
>(45-41)+1 //panjang kelas
```

```
5
```

```
>70+0.5 //tepi atas terbesar
```

```
70.5
```

```
>r=40.5:5:70.5; f=[10,12,18,34,20,6];
>T:=r[1:6]' | r[2:7]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

```
 tepi bawah tepi atas frekuensi
       40.5      45.5        10
       45.5      50.5        12
       50.5      55.5        18
       55.5      60.5        34
       60.5      65.5        20
       65.5      70.5         6
```

```
>(T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]); m=mean(t,f);
```

karena data tersebut merupakan data sampel, maka menggunakan rumus berikut

```
>sqrt(sum(f*(t-m)^2)/(sum(f)-1))
```

```
6.81649810861
```

Jadi, simpangan baku data kelompok tersebut adalah 6.81649810861

6.Mencari Jangkauan/Range

Jangkauan, atau biasa disebut range, merupakan perbedaan antara nilai data tertinggi dan nilai data terendah dalam suatu set data. Metode pencarian jangkauan berbeda antara data tunggal dan data kelompok.

a. Jangkauan/Range Data Tunggal

Bila ada sekumpulan data tunggal terurut dari yang terkecil sampai terbesar adalah

$$x_1, x_2, ..., x_n$$

maka jangkauannya adalah:

$$Jangkauan = x_n - n_1$$

Untuk menemukan jangkauan data tunggal di EMT dapat menggunakan perintah berikut:

> x=[data]; max(x)-min(x)

Contoh soal

Jangkauan dari data 30,60,87,55,87,98,22,75,81,70,69,84,75 adalah...

```
>x=[30,60,87,55,87,98,22,75,81,70,69,84,75]; max(x)- min(x)
```

```
76
```

Jadi, jangkauan dari data tersebut adalah 76

b. Jangkauan data kelompok

Jangkauan pada data berkelompok adalah selisih antara batas atas dari kelas tertinggi dengan batas bawah dari kelas terendah.

Pada EMT, untuk menemukan jangkauan dari data berkelompok dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas (Tb), panjang kelas (P), dan tepi atas kelas (Ta) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])

3. Menghitung jangkauan data berkelompok

> max(transpose(T[,2]))-min(transpose(T[,1]))
Contoh soal :
Berikut adalah data hasil dari pengukuran berat badan 20 siswa SD kelas V. Dari ke 20 siswa,siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 4 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 3 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 2 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 3 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 3 orang. Tentukan jangkauan dari
data hasil pengukuran berat badan 20 siswa di SD tersebut!
Jawab :

```
>printfile("Tabel jangkauan data kelompok.dat",7) //menyederhanakan informasi
```

```
Could not open the file
Tabel jangkauan data kelompok.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

```
>21-0.5 //tepi bawah terkecil
```

```
20.5
```

```
>(26-21)+1 //panjang kelas
```

```
6
```

```
>56+0.5 //tepi atas terbesar
```

```
56.5
```

```
>r=20.5:6:56.5; f=[5,4,3,2,3,3];
>T:=r[1:6]' | r[2:7]' | f'; writetable(T,labc=["tepi bawah","tepi atas","frekuensi"])
```

```
tepi bawah tepi atas frekuensi
      20.5      26.5         5
      26.5      32.5         4
      32.5      38.5         3
      38.5      44.5         2
      44.5      50.5         3
      50.5      56.5         3
```

```
>max(transpose(T[,2]))-min(transpose(T[,1]))
```

```
36
```

Jadi, Jangkauan dari data kelompok tersebut adalah 36

7.Menentukan ukuran letak

Ukuran letak merupakan ukuran untuk melihat dimana letak salah satu data dari sekumpulan banyak data yang ada. Yang termasuk ukuran ukuran letak antara lain adalah kuartil(Q), desil(D) dan persentil(P). Dalam menentukan ke-3 nya yang harus diingat adalah mengurutkan distribusi data dari yang terkecil sampai terbesar

1. Kuartil
Dalam EMT untuk menghitung kuartil bisa dilakukan dengan perintah
>quartiles(data)
perintah tersebut akan menghasilkan nilai Q1, Q2, Q3, nilai minimum dan nilai maksimum dari suatu data
2. Desil
Dalam EMT untuk menghitung desil bisa dilakukan dengan perintah
>quantile(data)
3. Persentil
Dalam EMT untuk menghitung persentil bisa dilakukan dengan perintah
>quantile(data)
perintah ">quantile(data)" dapat digunakan untuk menentukan desil dan persentil perbedaannya tergantung pada nilai dari pembaginya

Contoh soal
1. Tentukan Q1,Q2 dan Q3 dari data : 7,3,8,5,9,4,8,3,10,2,7,6,8,7,2,6,9.

```
>data=[7,3,8,5,9,4,8,3,10,2,7,6,8,7,2,6,9];
>urut=sort(data)
```

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

```
>quartiles(urut)
```

```
[2,  3.5,  7,  8,  10]
```

dari hasil di atad diperoleh nilai sebagai berikut :
Nilai minimal data = 2
Q1=3.5
Q2=7
Q3=8
Nilai maksimal data = 10

2. Tentukan D8 dari data : 6,3,8,9,5,9,9,7,5,7,4,5,8,3,7,6

```
>data=[6,3,8,9,5,9,9,7,5,7,4,5,8,3,7,6];
>urut=sort(data)
```

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

```
>quantile(urut,0.8)  //nilai 0.8 diapatkan karena kita akan mencari D8
```

```
8
```

Jadi, nilai dari D8 berdasarkan perhitungan di atas adalah 8

3.Tentukan persentil ke-65 dari data : 6,5,8,7,9,4,5,8,4,7,8,5,8,4,5

```
>data=[6,5,8,7,9,4,5,8,4,7,8,5,8,4,5];
>urut=sort(data)
```

```
[4,  4,  4,  5,  5,  5,  5,  6,  7,  7,  8,  8,  8,  8,  9]
```

```
>quantile(urut,65%)
```

```
7.1
```

NAMA : Dida Arkadia Ayu Jawata
NIM : 22305144005
KELAS : Matematika E 2023
Menggambar Grafik Statistika

Diagram Kotak

Diagram kotak atau box plot merupakan ringkasan distribusi sampel yang disajikan secara grafis yang bisa menggambarkan bentuk distribusi data (skewness), ukuran tendensi sentral dan ukuran penyebaran (keragaman) data pengamatan. Diagram kotak sering digunakan ketika jumlah distribusi data perlu dibandingkan. Diagram kotak menyajikan informasi tentang nilai–nilai inti dalam distribusi data termasuk juga pencilan. Pencilan adalah titik data yang terpaut jauh dari titik data lainnya.

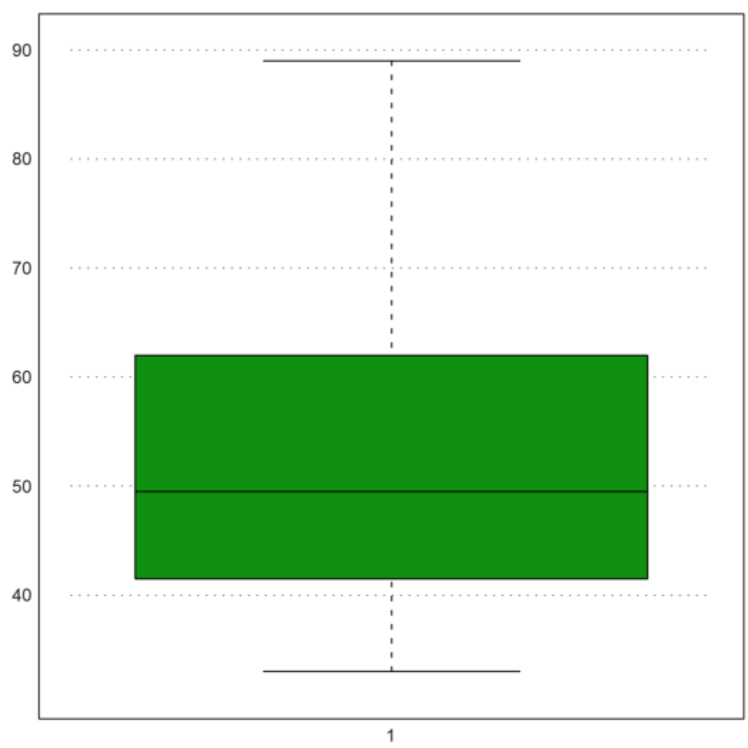
Contoh:
Diketahui data berat badan mahasiswa di Universitas A sebagai berikut.

```
>A=[55,50,33,42,44,37,63,74,56,34,51,43,45,39,64,77,60,35,53,43,48,41,65,87,61,36,54,44,49,41,66,89]
```

```
[55,  50,  33,  42,  44,  37,  63,  74,  56,  34,  51,  43,  45,  39,
64,  77,  60,  35,  53,  43,  48,  41,  65,  87,  61,  36,  54,  44,
49,  41,  66,  89]
```

Buatlah diagram kotak (box plot) kemudian tuliskan interpretasinya.

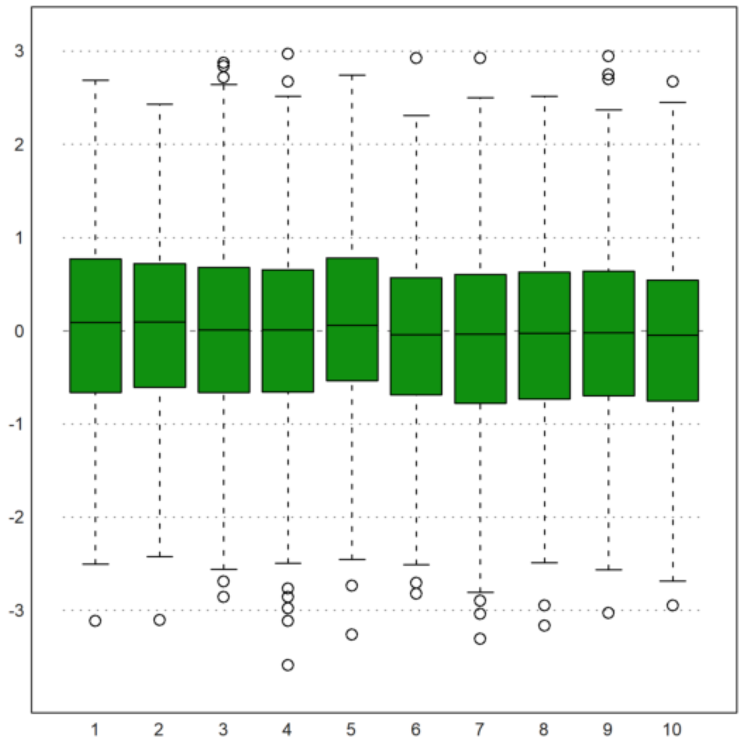
```
>boxplot(A):
```



Dari gambar box plot berat badan mahasiswa Universitas A, sepintas kita bisa menentukan beberapa ukuran statistik, meskipun tidak persis sekali. Nilai statistik pada badan boxplot berkisar pada: Nilai Minimum = 33 , Q1 = 41.5 , Median (Q2) = 49.5 , Q3 = 62 , Nilai Maksimum = 89 . Sebaran data tidak simetris, melainkan menjulur ke arah kanan (postively skewness). Karena nilai jarak Q1 dengan Q2 lebih pendek dari jarak Q2 dengan Q3, maka data lebih terpusat di kiri. Akan tetapi data tersebut tergolong cenderung mesokurtik karena jarak IQR dengan panjang hampir sama, dengan data berpusat di angka 49.5

Adapun contoh perbandingan 10 simulasi 500 nilai terdistribusi normal menggunakan box plot dan terdapat pencilan sebagai berikut.

```
> p=normal(10,500); boxplot(p):
```



pada diagram diatas, adalah membuat boxplot distribusi normal dengan rata-rata 10 dan standar deviasi 500. Boxplot adalah representasi grafis dari lokalitas, penyebaran, dan kecondongan sekelompok data numerik melalui kuartil mereka
2

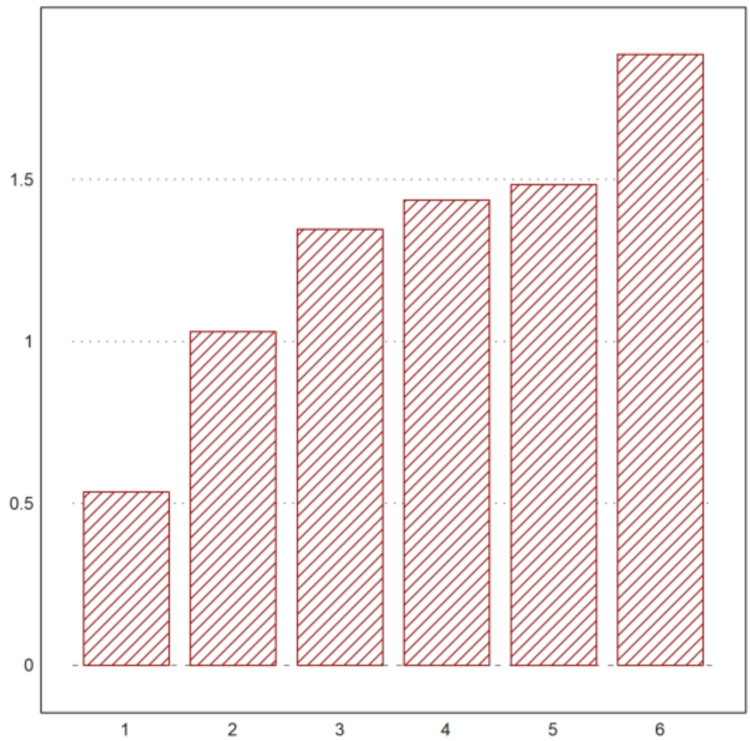
Diagram Batang

Diagram batang adalah representasi visual dari data yang menggunakan balok atau kolom vertikal untuk mewakili kategori, nilai atau variabel tertentu. Setiap kolom yang ada pada diagram batang memiliki frekuensi atau jumlah dalam kategori tersebut.

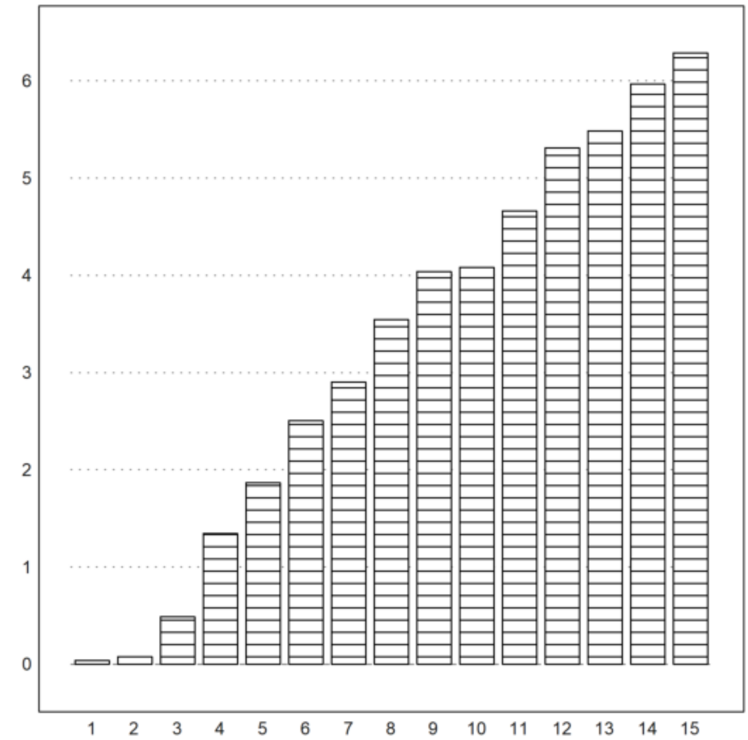
Contoh:

Kita akan membuat diagram batang secara random.

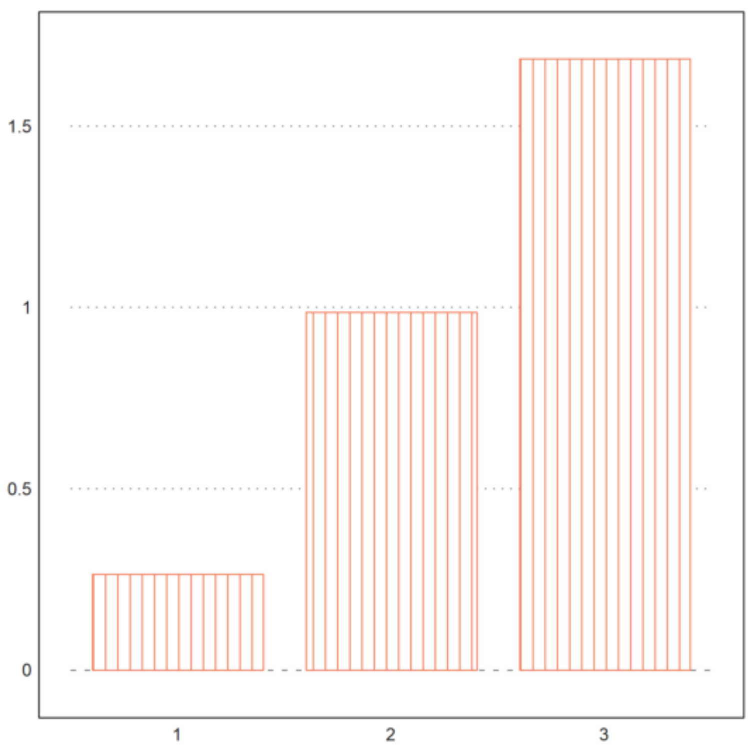
```
>columnsplot(cumsum(random(6)),style="/",color=red):
```




```
>columnsplot(cumsum(random(15)),style="-",color=black):
```

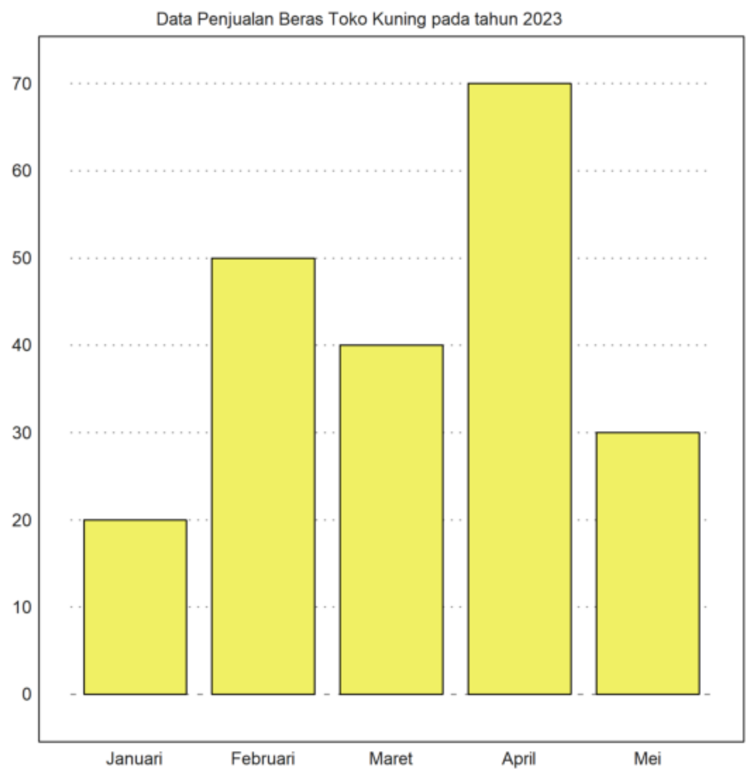



```
>columnsplot(cumsum(random(3)),style="|",color=orange):
```



Selanjutnya kita akan mencoba membuat diagram batang penjualan yang menggunakan variabel.

```
>months=["Januari","Februari","Maret","April","Mei"];
>values=[20,50,40,70,30];
>columnsplot(values,lab=months,color=yellow);
>title("Data Penjualan Beras Toko Kuning pada tahun 2023"):
```



Perintah ”columnsplot(values,lab=months,color=yellow);” merupakan sintaks untuk membuat diagram batang dengan menggunakan nilai dari variabel ”values”, label bulan dari variabel ”months”, dan warna kuning

Dari diagram batang tersebut kita bisa mengetahui data penjualan toko kuning selama lima bulan pada tahun 2023 yaitu, pada bulan Januari, Februari, Maret , April, Mei. Januari terjual 20 ton beras, Februari terjual 50 ton beras, Maret terjual 40 ton beras, April terjual 70 ton beras, dan Mei terjual 30 ton beras.

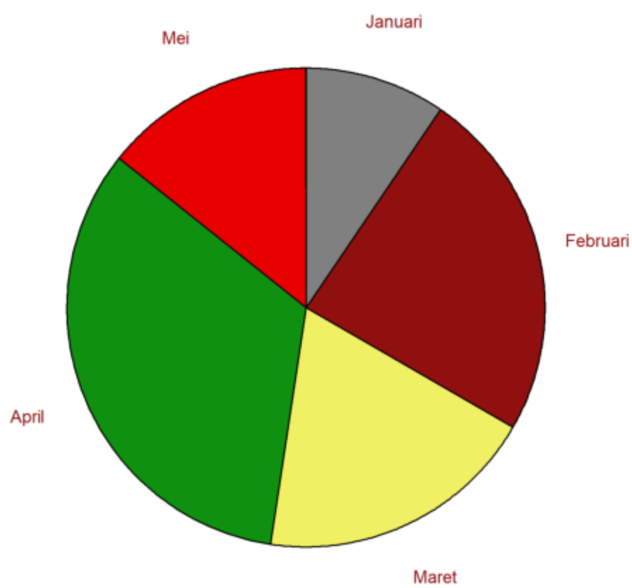
Diagram Lingkaran

Diagram lingkaran merupakan penyajian statistik data tunggal dalam bentuk lingkaran yang dibagi menjadi beberapa juring atau sektor yang menggambarkan banyak frekuensi untuk setiap data.Diagram lingkaran tidak menampilkan informasi frekuensi dari masing-masing data secara detail.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.9,0,0)]
```

```
[5.87532e+07,  2,  15,  3,  6.54049e+07]
```

```
>i=[1,2,3,4,5]; piechart(values[i],color=CP[i],lab=months[i]):
```

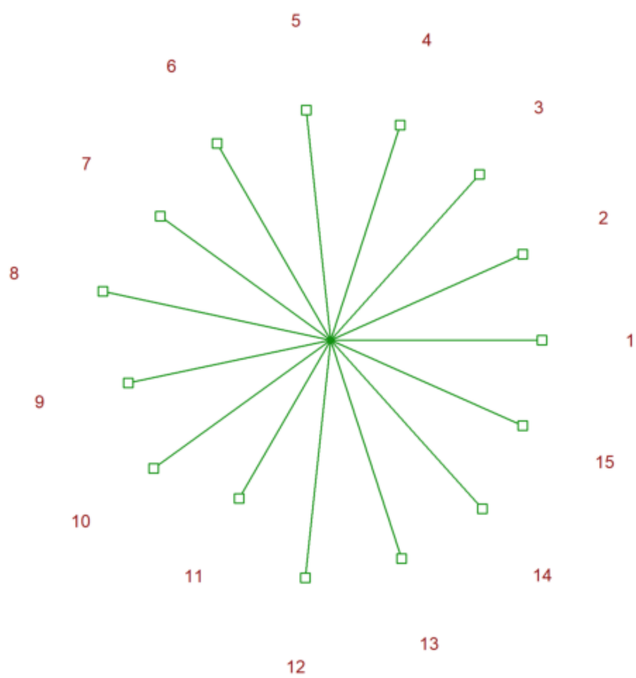


RGB adalah singkatan dari Red, Green, and Blue, dan setiap parameter mendefinisikan intensitas warna dengan nilai antara 0 dan 1. Warna pertama dalam daftar adalah warna abu-abu dengan jumlah merah, hijau, dan biru yang sama. Warna kedua merah, ketiga kuning, dan keempat hijau. Warna terakhir adalah warna merah dengan lebih banyak merah daripada hijau atau biru.

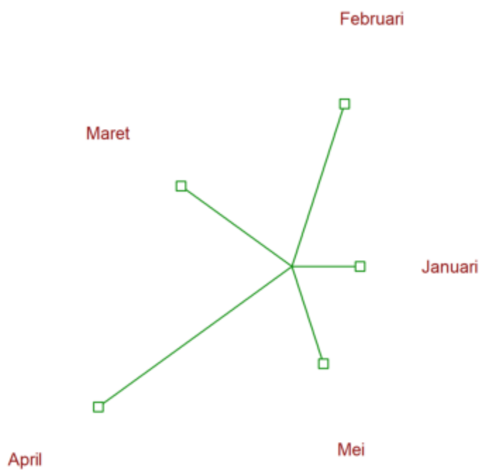
Diagram Bintang

Diagram bintang, terkadang disebut diagram radar atau diagram web, adalah metode perangkat grafis yang digunakan untuk menampilkan data multivariat. Multivariat dalam pengertian ini mengacu pada memiliki banyak karakteristik untuk diamati. Variabelnya juga harus berupa nilai yang berkisar.
Diagram bintang terdiri dari rangkaian jari-jari bersudut sama, yang disebut jari-jari, dengan masing-masing jari mewakili salah satu variabel. Panjang jari-jari data sebanding dengan besaran variabel pada titik data relatif terhadap besaran maksimum variabel di seluruh titik data.

```
>starplot(normal(1,15)+16,lab=1:15,>rays):
```




```
>starplot(values,lab=months,>rays):
```

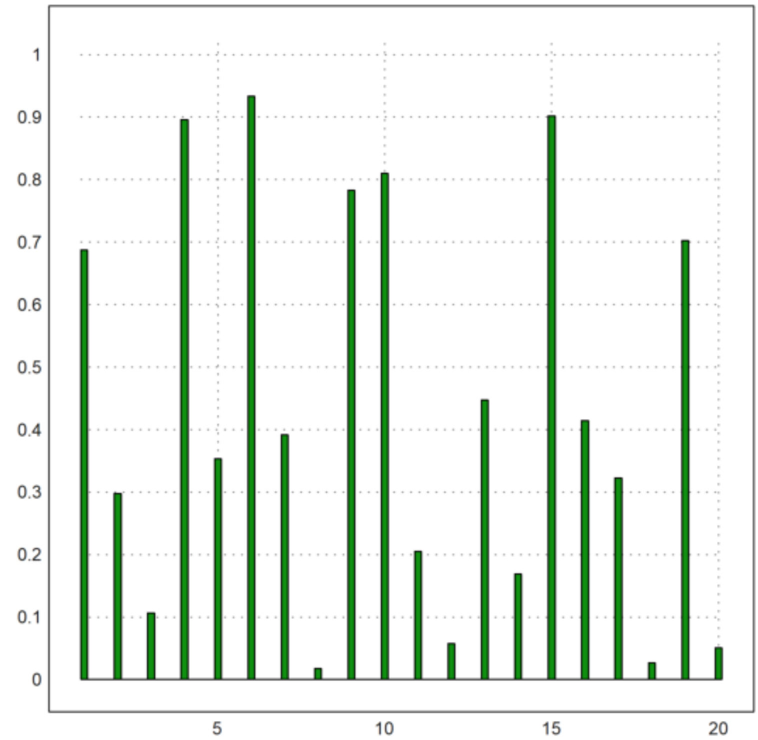


Syntax starplot(values,lab=months,rays) adalah perintah untuk membuat grafik bintang (star plot) dengan menggunakan nilai-nilai yang diberikan dalam vektor values, label sumbu yang diberikan dalam vektor months, dan jumlah rays yang menentukan jumlah garis radial yang digunakan dalam grafik

Diagram Impuls

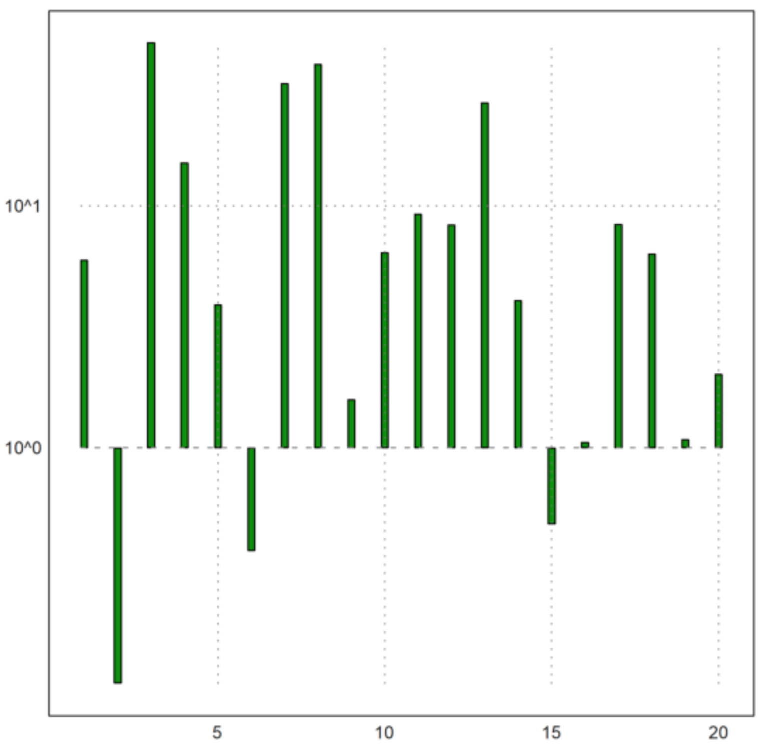
Impuls (impulse) adalah perubahan momentum. Contohnya adalah sebuah bola bermassa yang tengah ditendang, bola menggelinding yang dihentikan, bola jatuh yang memantul, mobil yang menabrak tembok, telur jatuh yang pecah.
Berikut adalah plot impuls dari data acak 1 sampai 20, terdistribusi secara merata di [0,1].

```
>plot2d(makeimpulse(1:20,random(1,20)),>bar):
```



Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
> logimpulseplot(1:20,-log(random(1,20))*10):
```

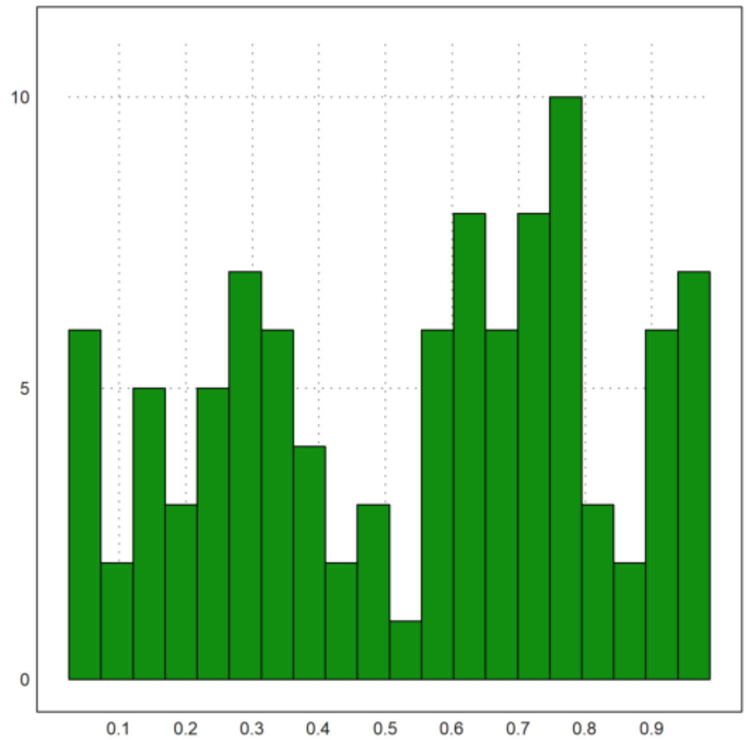


Jadi gambar grafiknya terlihat naik turun (mengalami perubahan).

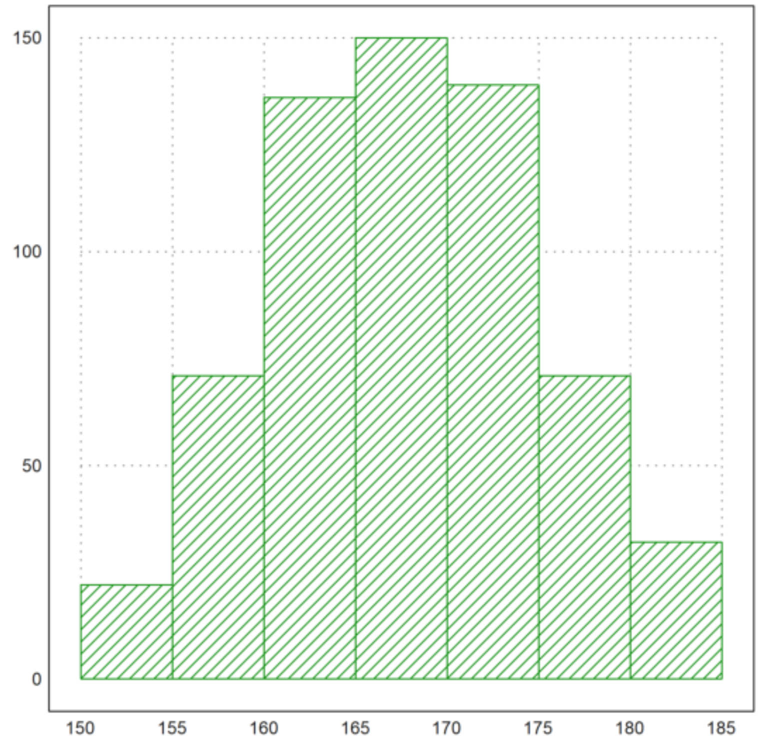
Histogram

Histogram adalah representasi grafis (diagram) yang mengatur dan menampilkan frekuensi data sampel pada rentang tertentu. Frekuensi data yang ada pada masing-masing kelas direpresentasikan dengan bentuk grafik diagram batang atau kolom.

```
>aspect(1); plot2d(random(100),>histogram):
```




```
>r=150:5:185; v=[22,71,136,150,139,71,32];
>plot2d(r,v,a=150,b=185,c=0,d=150,bar=1,style="/"):
```



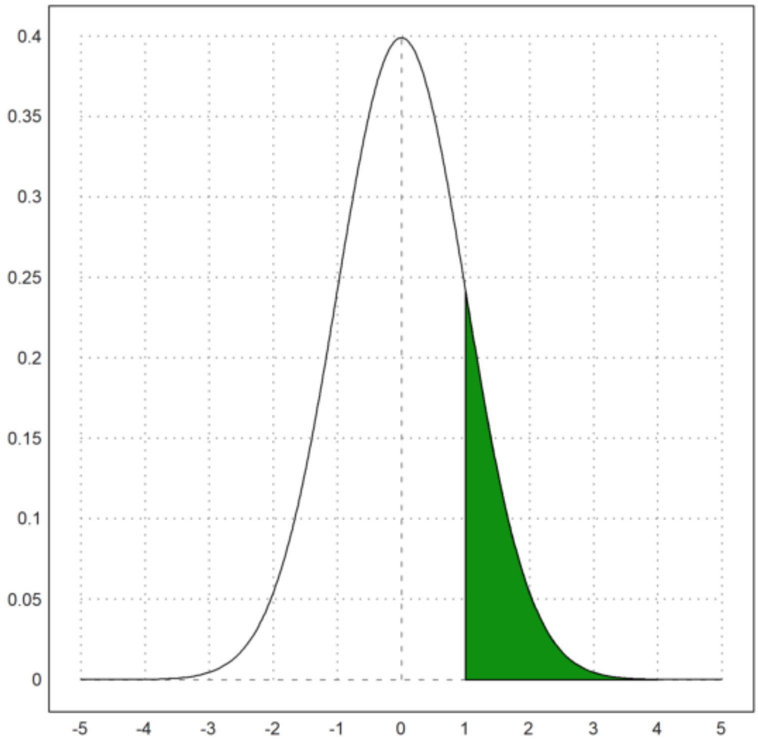
Pola ”r=150:5:185” berarti bahwa nilai r dimulai dari 150, kemudian bertambah 5 setiap kali, dan berakhir saat mencapai atau melebihi 185. Dengan pola ini, kita dapat menentukan nilai-nilai r yang sesuai.

Dari data yang diperoleh dapat diketahui bahwa dari rentang kelas 150-155 memiliki frekuensi 22, rentang kelas 155-160 memiliki frekuensi 71, dan seterusnya.

Kurva Fungsi Kerapatan Probabilitas

Secara teoritis kurva probabilitas populasi diwakili oleh poligon frekuensi relatif yang dimuluskan (variabel acak kontiniu diperlakukan seperti variabel acak diskrit yang rapat).Karena itu fungsi dari variabel acak kontiniu merupakan fungsi kepadatan probabilitas (probability density function – pdf). Pdf menggambarkan besarnya probabilitas per unit interval nilai variabel acaknya.

```
>plot2d("qnormal(x,0,1)",-5,5);  ...
>plot2d("qnormal(x,0,1)",a=1,b=4,>add,>filled):
```



Perintah "plot2d("qnormal(x,0,1)",-5,5)" digunakan untuk membuat plot dari distribusi normal dengan mean 0 dan standard deviation 1 di rentang -5 hingga 5

Probabilitas variabel acak x yang terletak antara 1 dan 4 memenuhi
P(1<X<4)= luas daerah hijau

Kurva Fungsi Distribusi Kumulatif

Cumulative Distribution Function (CDF) atau fungsi distribusi kumulatif adalah fungsi matematika yang digunakan untuk menghitung probabilitas variabel acak diskrit atau kontinu. CDF memberikan probabilitas bahwa variabel acak akan menghasilkan nilai kurang dari atau sama dengan nilai tertentu. Dalam hal ini, CDF dapat digunakan untuk menghitung probabilitas kumulatif dari variabel acak.

Berikut merupakan contoh kurva fungsi distribusi kumulatif kontinu:

```
>splot2d("normaldis",-3,5):
```

```
Function splot2d not found.
Try list ... to find functions!
Error in:
splot2d("normaldis",-3,5): ...
                          ^
```

Dapat kita lihat dalam kurva fungsi distribusi kumulatif kontinu terdiri atas tiga bagian yaitu:
1. Bernilai 0 untuk x di bawah minimal dari daerah rentang.
2. Merupakan fungsi monoton naik pada daerah rentang.
3. Mempunyai nilai konstan 1 di atas batas maksimum daerah rentangnya.

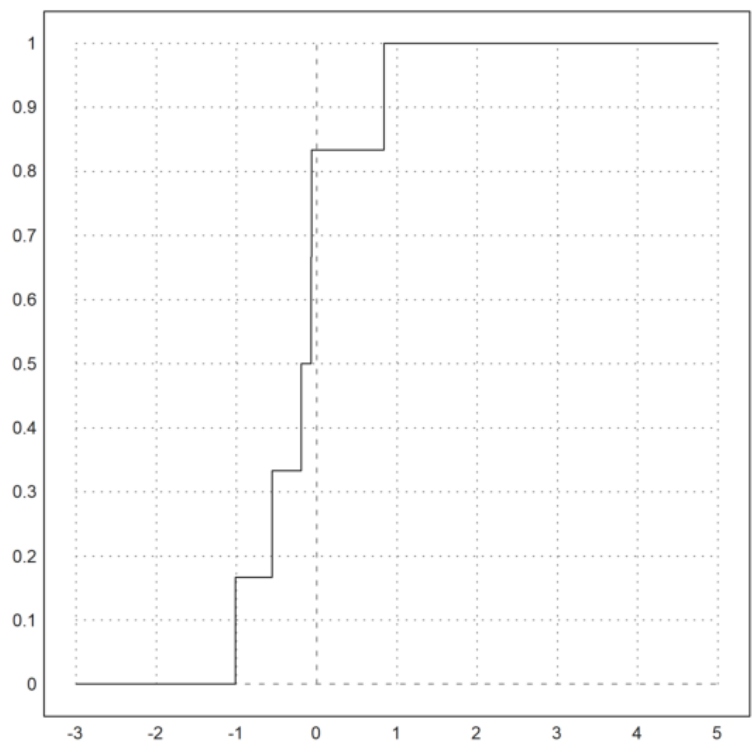
Adapun contoh kurva fungsi distribusi kumulatif diskrit sebagai berikut.

```
>x=normal(1,6);
```

Baris kode tersebut akan menghasilkan suatu nilai acak dari distribusi normal dengan mean 1 dan deviasi standar 6, dan nilai tersebut disimpan dalam variabel x. Variabel x kemudian dapat digunakan dalam perhitungan atau analisis selanjutnya

Fungsi empdist(x,vs) membutuhkan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

```
>xs=sort(x);
>plot2d("empdist",-3,5;xs):
```



Grafik fungsi distribusi kumulatif peubah acak diskrit merupakan fungsi tangga naik dengan nilai terendah 0 dan nilai tertinggi 1.

Nama : Oktavia Kusuma Wardhani
NIM : 22305144013
Kelas : Matematika E

Cakupan Materi 1) Diagram Titik
2) Diagram Garis
3) Kurva Regresi
4) Menampilkan Tabel Data Frame

Diagram Titik Diagram titik atau bisa disebut Scatter Plot adalah tipe grafik yang digunakan untuk menampilkan nilai-nilai dua variabel pada sumbu horizontal dan vertikal. Setiap titik dalam diagram mewakili satu observasi atau data point. Diagram titik sangat berguna untuk menemukan pola atau hubungan antara dua variabel, serta untuk mengevaluasi distribusi data.

Dalam scatter plot, sumbu horizontal umumnya digunakan untuk variabel independen, sementara sumbu vertikal digunakan untuk variabel dependen. Dengan melihat pola penyebaran titik-titik, kita akan mendapatkan wawasan tentang apakah ada korelasi antara dua variabel dan jenis korelasi apa yang mungkin ada (positif, negatif, atau tidak ada korelasi).

```
>x=normal(1,150); plot2d(x,x+rotright(x),>points,style=".."):
```

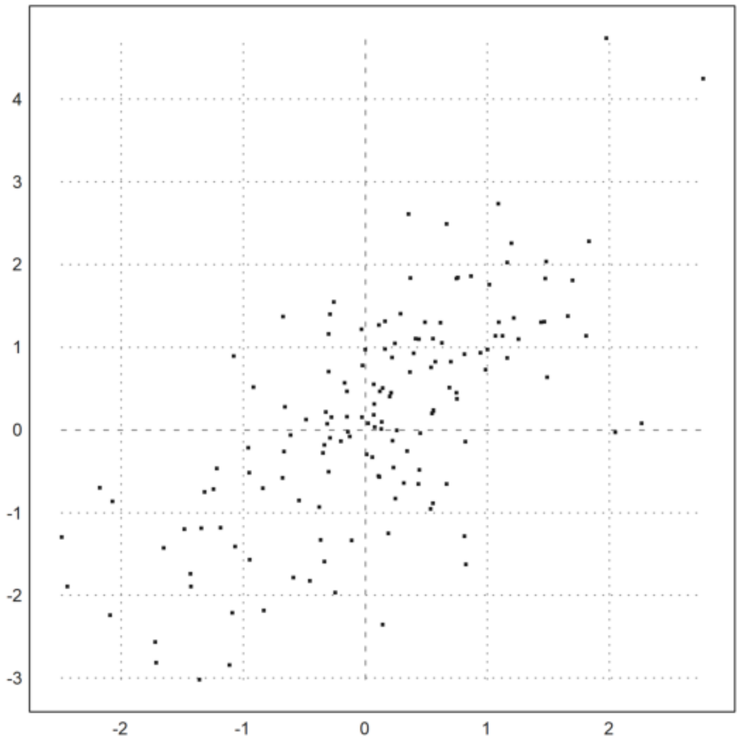


Diagram titik di atas memvisualisasikan data yang dihasilkan oleh fungsi normal dengan parameter mean 1 dan standar deviasi 150. Diagram di atas juga menunjukkan adanya korelasi yang positif karena pergeseran data tersebut ke kanan.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- x=normal(1,150) menghasilkan data dari distribusi normal dengan mean 1 dan standar deviasi 150.
- plot2d(x,x+rotright(x),>points,style="..") digunakan untuk membuat plot 2D dari data x dan x+rotright(x) sebagai sumbu x dan y. >points digunakan untuk menunjukkan bahwa plot yang dihasilkan berupa titik-titik, dan style=".." menentukan gaya dari titik-titik tersebut.

```
>plot2d(normal(1500),normal(1500),>points,grid=6,style=".."):
```

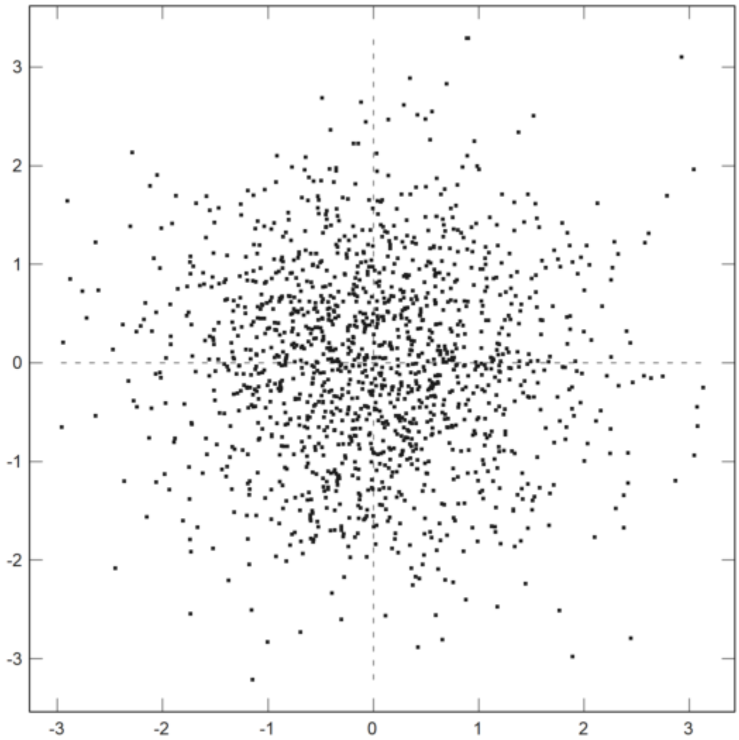


Diagram titik di atas memvisualisasikan distribusi dari dua set data yang dihasilkan oleh distribusi normal.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
plot2d(normal(1500),normal(1500),>points,grid=6,style="..") digunakan untuk membuat plot 2D dari dua distribusi normal yang menghasilkan masing-masing 1500 data. >points menunjukkan bahwa plot yang dihasilkan berupa titik-titik, grid=6 menentukan ukuran grid, dan style=".." menentukan gaya dari titik-titik tersebut.

```
>{MS,hd}:=readtable("table1.dat",tok2:=["m","f"]); ...
>writetable(MS,labc=hd,tok2:=["m","f"]);
```

```
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;
```

Tabel di atas memvisualisasikan data yang berisi mengenai survei anak, jenis kelamin mereka, usia mereka, usia orang tua mereka dan jumlah saudara kandung mereka.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- {MS,hd}:=readtable("table1.dat",tok2:=["m","f"]) digunakan untuk membaca data dari file "table1.dat" dan menyimpannya dalam variabel MS dan hd. tok2:=["m","f"] menunjukkan bahwa data dalam file tersebut memiliki format "m" dan "f".
- writetable(MS,labc=hd,tok2:=["m","f"]) digunakan untuk menulis kembali data ke file dengan menggunakan label hd sebagai nama kolom dan format data "m" dan "f".

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

```
Variable or function MS not found.
Error in:
scatterplots(tablecol(MS,3:5),hd[3:5]): ...
                         ^
```

Diagram titik di atas memvisualisasikan hubungan antara tiga variabel dalam tabel data "MS", yaitu usia anak, ibu, dan ayah.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- scatterplots(tablecol(MS,3:5),hd[3:5]) digunakan untuk membuat scatter plot (diagram titik) dari tiga kolom data dalam tabel MS dengan menggunakan label hd sebagai nama kolom.
- tablecol(MS,3:5) digunakan untuk memilih tiga kolom data dari tabel MS.
- hd[3:5] digunakan untuk memilih tiga label kolom dari hd.

Contoh Soal

Tabel berikut ini memberikan informasi mengenai kandungan gula (gram) dan jumlah kalori dalam satu sajian dari 13 sampel merek sereal.

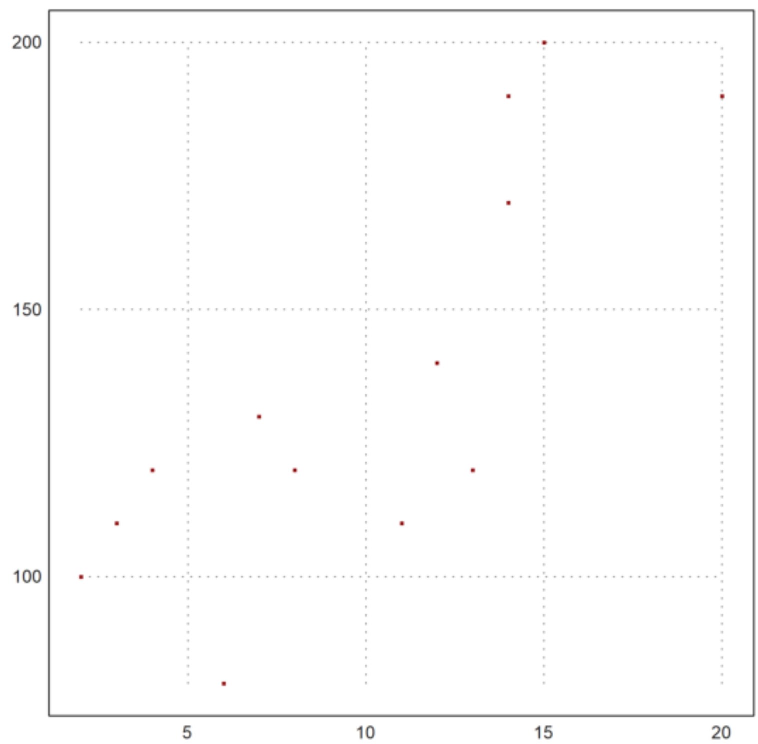
```
>DATA:=[ ...
>1,4,120; ...
>2,15,200; ...
>3,12,140; ...
>4,11,110; ...
>5,8,120; ...
>6,6,80; ...
>7,14,170; ...
>8,2,100; ...
>9,7,130; ...
>10,14,190; ...
>11,20,190; ...
>12,3,110; ...
>13,13,120];
>VAR:= ["MERK", "GULA(gram)", "KALORI"];
>writetable(DATA,labc=VAR);
```

MERK	GULA(gram)	KALORI
1	4	120
2	15	200
3	12	140
4	11	110
5	8	120
6	6	80
7	14	170
8	2	100
9	7	130
10	14	190
11	20	190
12	3	110
13	13	120

a. Gambarkan diagram titik atau scatter plot dari data di atas.
b. Bagaimana pola penyebaran titik-titik yang telah digambar pada diagram di atas?
c. Kesimpulan seperti apa yang dapat kalian ambil mengenai hubungan antara gula (gram) dan jumlah kalori?

Penyelesaian:

```
>GULA:=[4,15,12,11,8,6,14,2,7,14,20,3,13];
>KALORI:=[120,200,140,110,120,80,170,100,130,190,190,110,120];
>plot2d(GULA,KALORI,>points,color=red,style=".."):
```



a. Diagram titik atau scatter plot di atas merupakan hasil visualisasi dari tabel data yang telah disebutkan sebelumnya.
b. Pada diagram di atas dapat terlihat bahwa pola penyebaran titiknya mempunyai kecenderungan semakin naik ke atas jika dilihat dari kiri bawah ke kanan atas.
c. Kesimpulan yang dapat diambil yaitu semakin tinggi kandungan gula maka semakin tinggi jumlah kalorinya.

Diagram Garis Diagram garis atau bisa disebut line chart adalah jenis grafik statistik yang digunakan untuk menunjukkan perubahan atau tren sepanjang waktu atau variabel independen lainnya. Ini adalah salah satu cara yang paling umum digunakan untuk memvisualisasikan data berurutan dalam statistika.

Dalam diagram garis, data diplotkan sebagai titik-titik dan kemudian dihubungkan dengan garis lurus. Ini membantu untuk melihat perubahan atau tren dari satu titik ke titik berikutnya dengan jelas. Diagram garis sangat berguna untuk menyoroti pola, tren, atau fluktuasi dalam data sepanjang periode waktu tertentu.

Contoh umum penggunaan diagram garis melibatkan waktu di sumbu horizontal (x) dan nilai atau frekuensi di sumbu vertikal (y). Misalnya, diagram garis dapat digunakan untuk menunjukkan perubahan suhu sepanjang waktu, pertumbuhan populasi, atau kinerja keuangan perusahaan sepanjang beberapa kuartal.

Untuk memetakan data, kita mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam jumlah kursi.

```
>BW := [...
>1990,662,319,239,79,8,17; ...
>1994,672,294,252,47,49,30; ...
>1998,669,245,298,43,47,36; ...
>2002,603,248,251,47,55,2; ...
>2005,614,226,222,61,51,54; ...
>2009,622,239,146,93,68,76; ...
>2013,631,311,193,0,63,64];
```

Sintaks tersebut digunakan untuk membuat matriks BW yang berisi data-data tahun dan jumlah kursi untuk beberapa partai politik pada setiap tahun.

```
>P:=["CDU/CSU","SPD","FDP","Gr","Li"];
```

Sintaks tersebut digunakan untuk membuat array P yang berisi label-label untuk setiap kolom pada matriks BW.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- BT:=BW[,3:7]; BT:=BT/sum(BT) digunakan untuk mengambil kolom ke-3 hingga ke-7 dari matriks BW, kemudian membagi setiap nilai dalam matriks tersebut dengan jumlah total nilai dalam matriks tersebut. Kolom 3 sampai 7 adalah jumlah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi seluruhnya. Sedangkan kolom 1 adalah tahun pemilihan.
- YT:=BW[,1]' digunakan untuk mengambil transpose dari kolom pertama matriks BW dan menyimpannya dalam variabel YT.

```
>writetable(BT*100,wc=6,dc=0,>fixed,labc=P,labr=YT)
```

```
        CDU/CSU   SPD   FDP    Gr    Li
    1990     48    36    12     1     3
    1994     44    38     7     7     4
    1998     37    45     6     7     5
    2002     41    42     8     9     0
    2005     37    36    10     8     9
    2009     38    23    15    11    12
    2013     49    31     0    10    10
```

Sintaks tersebut digunakan untuk menulis matriks BT yang telah dikalikan dengan 100 ke dalam sebuah file. Parameter wc=6 menunjukkan lebar atau sekat kolom, dc=0 menunjukkan jumlah digit di belakang koma, >fixed menunjukkan bahwa jumlah digit di belakang koma tetap, labc=P menunjukkan label untuk kolom, dan labr=YT menunjukkan label untuk baris.

```
>BT1:=(BT.[1;1;0;0;0])'*100
```

```
  [84.29,  81.25,  81.1659,  82.7529,  72.9642,  61.8971,  79.8732]
```

Sintaks di atas digunakan untuk mengalikan baris pertama dari matriks BT dengan 100 dan menyimpan hasilnya dalam variabel BT1. Angka-angka dalam tanda kurung siku menunjukkan indeks baris yang ingin diambil dari matriks BT (pada sintaks tersebut indeks baris yang ingin diambil yaitu baris pertama), sedangkan tanda kutip di sebelah kanan menunjukkan bahwa kita ingin mengalikan baris tersebut dengan 100.

```
>statplot(YT,BT1,"b"):
```

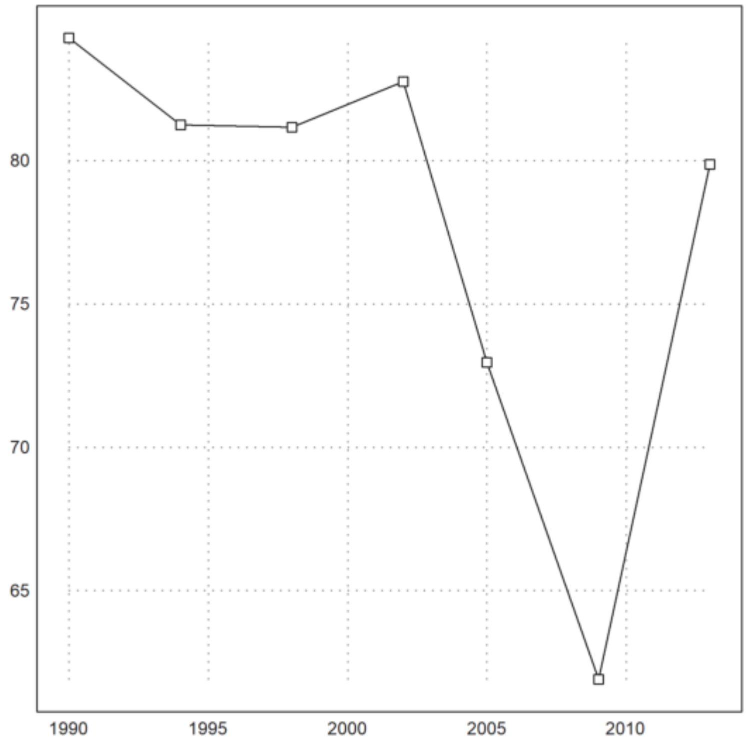


Diagram di atas memvisualisasikan persentase perolehan jumlah kursi setiap partai di setiap tahunnya, yaitu dari tahun 1990-2013.

Tipe plotplot yang tersedia adalah sebagai berikut:
- 'p' : plot titik
- 'l' : plot garis
- 'b' : keduanya (titik dan garis)
- 'h' : alur histogram
- 's' : plot permukaan

Contoh Soal

Perhatikan tabel di bawah ini!

```
>BARANG:=[ ...
>2015,13000,10000,2700,10700; ...
>2020,15000,12500,4000,12300];
>VRB := ["TAHUN","SUSU(kaleng)","GULA(kg)","JAGUNG(kg)","BERAS(kg)"];
>writetable(BARANG,labc=VRB);
```

```
TAHUN SUSU(kaleng)  GULA(kg) JAGUNG(kg) BERAS(kg)
 2015        13000     10000       2700     10700
 2020        15000     12500       4000     12300
```

Hitunglah persentase kenaikan harga dari tahun 2015 hingga tahun 2020 (jika penghitungan indeks harga menggunakan metode agregatif sedehana) dan gambarkan grafik kenaikan harganya menggunakan diagram garis!

Penyelesaian:

```
>HARGA:=BARANG[,2:5]; TAHUN:=BARANG[,1]';
>Po:=(13000+10000+2700+10700) // total harga barang di tahun 2015
```

```
36400
```

```
>Pn:=(15000+12500+4000+12300) // total harga barang di tahun 2020
```

```
43800
```

```
>Persentase:=(Pn/Po)*100
```

```
120.32967033
```

```
>Kenaikan:=(120.3-100)
```

```
20.3
```

Berdasarkan penghitungan tersebut, dapat disimpulkan bahwa dari tahun 2015 ke tahun 2020 terdapat kenaikan harga sebesar 20,3%.

```
>P1:=[36400,43800];
>statplot(TAHUN,P1,"b"):
```

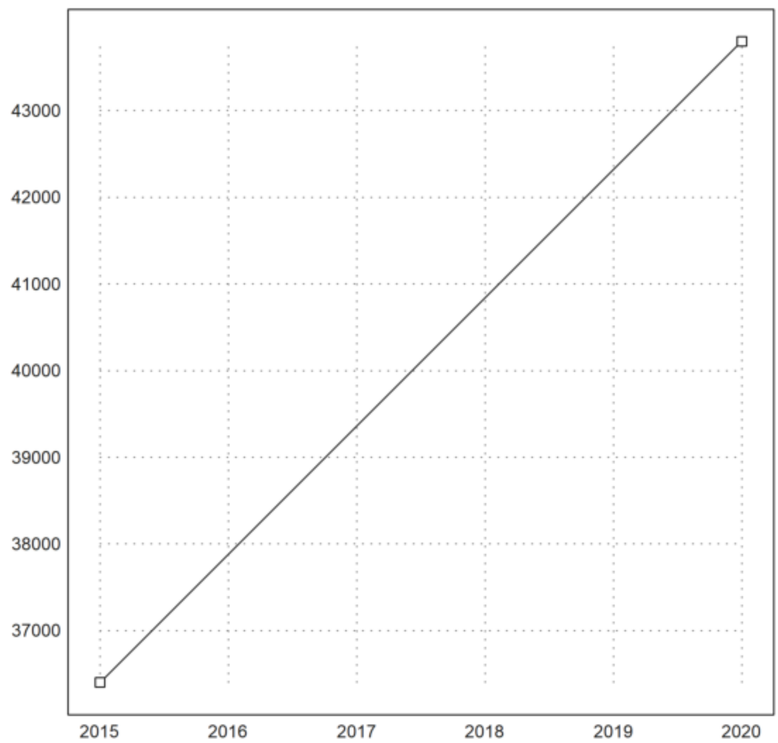


Diagram garis di atas memvisualisasikan kenaikan harga barang dari tahun 2015 hingga tahun 2020.

Kurva Regresi Kurva regresi adalah representasi grafis dari model regresi yang digunakan untuk memodelkan hubungan antara variabel independen (X) dan variabel dependen (Y). Persamaan regresi dapat digunakan untuk memprediksi atau mengestimasi nilai dari variabel tak bebas berdasarkan informasi dari variabel bebas. Persamaan regresi linear merupakan suatu persamaan yang berupa garis lurus, sedangkan persamaan regresi nonlinear bukan merupakan persamaan garis lurus.

Persamaan regresi linier umumnya ditulis sebagai Y=a+bX, di mana:
Y = garis regresi/ variable response
a = konstanta (intercept), perpotongan dengan sumbu-y
b = konstanta regresi (slope), kemiringan
X = variabel bebas/ predictor

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi fit. Sebagai permulaan, kami 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
```

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- x=1:10; y=[2,3,1,5,6,3,7,8,9,8];: Perintah ini digunakan untuk membuat vektor x yang berisi bilangan bulat dari 1 hingga 10, dan vektor y yang berisi nilai-nilai acak.
- writetable(x'|y',labc=["x","y"]): Perintah ini digunakan untuk membuat tabel dua kolom dari dua vektor x dan y. Parameter x'|y' menunjukkan bahwa vektor x dan y akan disusun secara vertikal, sedangkan labc=["x","y"] mengatur label kolom tabel.

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. Pertama koefisien kecocokan linier.

```
>p=polyfit(x,y,1)
```

```
[0.733333,  0.812121]
```

Sintaks di atas digunakan untuk melakukan fitting kurva polinomial orde satu pada data yang diberikan. Parameter x dan y adalah data yang akan di-fit, sedangkan 1 menunjukkan orde polinomial yang digunakan. Hasil fitting akan disimpan dalam variabel p.

Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

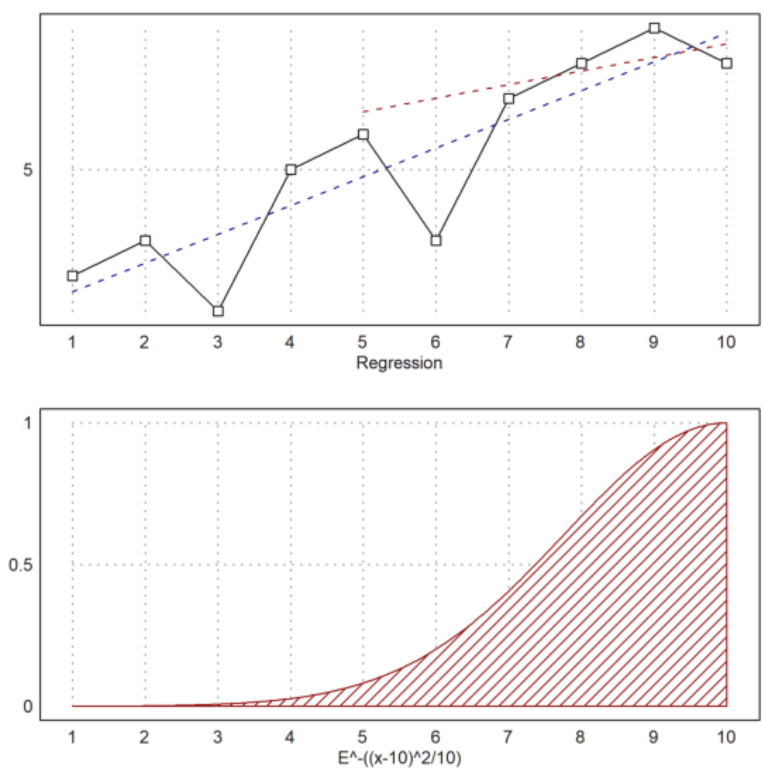
```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566,  0.38319]
```

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- w &= "exp(-(x-10)ˆ2/10)": digunakan untuk mendefinisikan fungsi berat (weight function) w sebagai exp(-(x-10)ˆ2/10). Fungsi berat digunakan dalam fitting polinomial untuk memberikan bobot pada setiap titik data, sehingga titik-titik data yang lebih dekat dengan nilai 10 akan diberi bobot yang lebih tinggi.
- pw=polyfit(x,y,1,w=w(x)): digunakan untuk melakukan fitting polinomial pada data yang diberikan dengan menggunakan fungsi berat w. Fungsi polyfit digunakan untuk menemukan koefisien polinomial yang sesuai dengan data. Dalam kasus ini, fitting dilakukan dengan polinomial derajat 1, dengan menggunakan fungsi berat yang telah didefinisikan sebelumnya.

Kami memasukkan semuanya ke dalam satu plot untuk titik dan garis regresi, dan untuk bobot yang digunakan.

```
>figure(2,1);  ...
>figure(1); statplot(x,y,"b",xl="Regression"); ...
>  plot2d("evalpoly(x,p)",>add,color=blue,style="--"); ...
>  plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"); ...
>figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w); ...
>figure(0):
```



Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- figure(2,1); digunakan untuk mengatur tampilan grafik dengan berbeda nomor tampilan (handle).
- statplot(x,y,"b",xl="Regression"): digunakan untuk menampilkan data (x,y) dengan label nama sumbu-x nya yaitu "Regression".
- plot2d("evalpoly(x,p)",>add,color=blue,style="–"): fungsi evalpoly digunakan untuk menghitung nilai polinomial pada titik-titik data yang diberikan. Dalam hal ini, evalpoly(x,p) menghitung nilai polinomial pada titik-titik data yang diberikan dari polinomial pertama (p). Hasilnya kemudian digunakan untuk menambahkan plot polinomial pertama ke grafik dengan warna biru dan gaya putus-putus.
- plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="–"): fungsi evalpoly digunakan untuk menghitung nilai polinomial yang diperoleh dari polyfit dengan fungsi berat. Dalam hal ini, evalpoly(x,pw) menghitung nilai polinomial dengan fungsi berat pada titik-titik data dari 5 hingga 10. Hasilnya kemudian digunakan untuk menambahkan plot polinomial dengan fungsi berat ke grafik dengan warna merah dan gaya putus-putus.
- plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w): sintaks tersebut digunakan untuk membuat plot dari area di bawah kurva polinomial dengan fungsi berat w pada rentang 1 hingga 10, dengan gaya pengisian warna merah dan label sumbu x yang diberi nilai w

---

Menampilkan Data Frame Data frame biasanya merujuk pada struktur data tabular dua dimensi yang digunakan untuk menyimpan dan mengorganisir data. Data frame adalah konsep yang umumnya terkait dengan pemrograman statistik, terutama dalam bahasa seperti R dan Python (menggunakan pustaka pandas).

Di direktori buku catatan ini kita akan menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file tersebut. Datanya berasal dari buku online Jerman "Einführung in die Statistik mit R" yang dibuat oleh A. Handl.

```
>printfile("table.dat",4);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

Sintaks tersebut digunakan untuk mencetak isi file "table.dat" pada empat baris pertama.

Tabel berisi 7 kolom angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan terjemahan kami sendiri untuk token.

Untuk ini, kami mendefinisikan set token. Fungsi strtokens() mendapatkan vektor string token dari string yang diberikan.

```
>mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
```

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- mf:=["m","f"]: Mendefinisikan variabel mf sebagai array yang berisi dua elemen, yaitu "m" dan "f". Kemungkinan ini digunakan untuk merepresentasikan jenis kelamin, di mana "m" mewakili laki-laki dan "f" mewakili perempuan.
- yn:=["y","n"]: Mendefinisikan variabel yn sebagai array yang berisi dua elemen, yaitu "y" dan "n". Kemungkinan ini digunakan untuk merepresentasikan jawaban ya ("y") dan tidak ("n") dari suatu pertanyaan.
- ev:=strtokens("g vg m b vb"): Mendefinisikan variabel ev sebagai hasil dari pemisahan string "g vg m b vb" berdasarkan spasi. Kemungkinan ini digunakan untuk merepresentasikan kategori tertentu yang terkait dengan data atau survei.

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen tok2, tok4 dll. adalah terjemahan dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda harus menyediakannya dengan ":=".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
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;
```

Sintaks ini digunakan untuk membaca data dari file "table.dat" ke dalam sebuah tabel. Pada saat membaca file, argumen opsional tok2, tok4, tok5, dan tok7 digunakan untuk menentukan bagaimana data dalam file tersebut akan diinterpretasikan.

```
>load over statistics;
```

Sintaks ini digunakan untuk memuat data atau variabel yang telah disimpan sebelumnya.

Untuk mencetak, kita perlu menentukan set token yang sama. Kami mencetak empat baris pertama saja.

```
>writetable(MT[1:4],labc=hd,wc=6,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
MT is not a variable!
Error in:
writetable(MT[1:4],labc=hd,wc=6,tok2:=mf,tok4:=yn,tok5:=ev,tok ...
                  ^
```

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan terlebih dahulu, kita hanya perlu menentukan, kolom mana yang berisi token dan bukan angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Fungsi readtable() sekarang mengembalikan satu set token.

Berikut adalah penjelasan dari setiap perintah dalam sintaks tersebut:
- ctok=[2,4,5,7]: digunakan untuk mendefinisikan variabel ctok sebagai array yang berisi angka 2, 4, 5, dan 7. Kemungkinan ini digunakan untuk menentukan kolom-kolom tertentu yang akan dibaca dari file "table.dat".
- {MT,hd,tok}=readtable("table.dat",ctok=ctok): sintaks ini menggunakan fungsi readtable untuk membaca data dari file "table.dat" ke dalam tiga variabel yang berbeda, yaitu MT, hd, dan tok, dengan menggunakan kolom-kolom yang telah ditentukan sebelumnya dalam variabel ctok.

```
>tok
```

```
Variable tok not found!
Error in:
tok ...
   ^
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke angka.

String khusus NA="." ditafsirkan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
>MT[1]
```

```
MT is not a variable!
Error in:
MT[1] ...
     ^
```

Sintaks tersebut digunakan untuk mengakses elemen pertama dari variabel MT yang telah diimpor sebelumnya. Berikut isi tabel dengan nomor yang belum diterjemahkan.

```
>writetable(MT,wc=5)
```

```
Variable or function MT not found.
Error in:
writetable(MT,wc=5) ...
             ^
```

Untuk kenyamanan, Anda dapat memasukkan keluaran readtable() ke dalam list.

```
>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]); ...
                 ^
```

Sintaks ini digunakan untuk mengekstrak kolom ke-5 dan ke-6 dari matriks atau tabel MT. Hasilnya akan disimpan dalam variabel c dan i secara berturut-turut. Dengan kata lain, c akan berisi kolom ke-5 dari MT, sedangkan i akan berisi kolom ke-6 dari MT.

Kita bisa menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

```
MT is not a variable!
Error in:
j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok) ...
                             ^
```

Berikut adalah maksud dari sintaks tersebut:
- MT[i,j]: Mengambil subset dari tabel MT yang terdiri dari baris-baris yang ditentukan oleh i dan kolom-kolom yang ditentukan oleh j. Ini mungkin digunakan untuk membuat subset dari data yang akan ditulis ke dalam file.
- labc=hd[j]: Menentukan label kolom untuk subset kolom yang dipilih dari MT berdasarkan label kolom dari hd. Ini mungkin digunakan untuk menetapkan label yang sesuai untuk kolom-kolom yang akan ditulis ke dalam file.
- ctok=[2]: Menentukan token tertentu yang akan digunakan saat menulis tabel ke dalam file. Ini mungkin digunakan untuk menetapkan cara khusus untuk menginterpretasikan data saat ditulis ke dalam file.
- tok=tok: Menggunakan token yang telah ditentukan sebelumnya saat menulis tabel ke dalam file. Ini mungkin digunakan untuk memastikan bahwa token yang sama digunakan saat menulis dan membaca tabel.

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Tabel dalam kasus ini.

```
>MT=Table[1];
```

```
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.

```
>{xu,count}=getstatistics(tablecol(MT,5)); xu, count,
```

```
Variable or function MT not found.
Error in:
{xu,count}=getstatistics(tablecol(MT,5)); xu, count, ...
                                    ^
```

Variabel xu adalah elemen unik (indeks) dari kolom ke-5 tabel MT, sedangkan variabel count adalah jumlah data dari variabel xu.

Kita bisa mencetak hasilnya di tabel baru.

```
>writetable(count',labr=tok[xu])
```

```
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])',ctok ...
               ^
```

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);
>
```

Nama : Fransisca Renita Pejoresa
NIM : 22305144012
Kelas : Matematika E 2022

Sub Topik 8 : Perhitungan untuk Uji Statistika

Dalam Euler, banyak uji yang diterapkan. Semua uji dalam Euler mengembalikan kesalahan yang diterima jika hipotesis nol ditolak.

Setiap jenis uji statistik memiliki asumsi dan syarat tertentu yang harus dipenuhi sebelum dapat diterapkan pada data. Pemilihan jenis uji statistik yang tepat sangat penting untuk memastikan hasil analisis data yang akurat dan dapat diandalkan.

Simulasi Monte Carlo

Simulasi Monte Carlo yang digunakan untuk memecahkan masalah dengan menggunakan sampel acak berulang untuk memperkirakan distribusi dari suatu fenomena. Dalam konteks uji statistika, Monte Carlo dapat digunakan untuk menghitung nilai-nilai yang sulit atau tidak mungkin dihitung secara analitik, seperti p-value dalam uji chi-square.

Euler dapat digunakan untuk mensimulasikan kejadian acak. Berikut ini contoh yang mensimulasikan 1000 kali 3 lemparan dadu, dan ditampilkan distribusi jumlahnya.

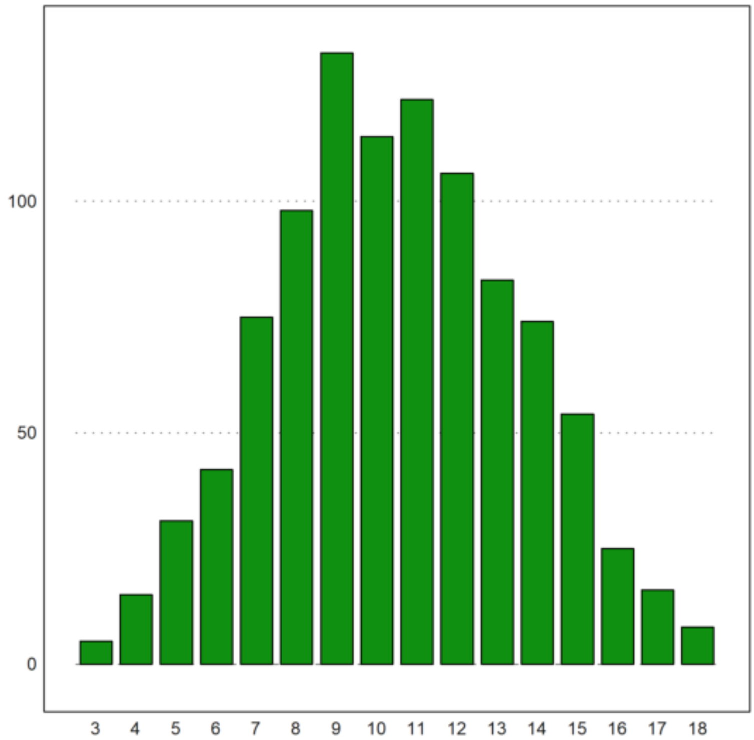
Fungsi ini akan menghasilkan 1000 bilangan bulat acak antara 3 dan 6, menjumlahkannya, dan menyimpan hasilnya dalam ds.

Fungsi getmultiplicities(v,x) mengembalikan kelipatan dari nilai-nilai dalam v dalam vektor x.
Dalam hal ini, fs akan berisi kelipatan dari nilai 3 hingga 18 dalam vektor ds.

```
>ds:=sum(intrandom(1000,3,6))'; fs=getmultiplicities(3:18,ds)
```

```
[5,  15,  31,  42,  75,  98,  132,  114,  122,  106,  83,  74,  54,
25,  16,  8]
```

```
>columnsplot(fs,lab=3:18):
```



Untuk menentukan distribusi yang diharapkan tidak begitu mudah. Diperlukan rekursi lanjutan untuk hal ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat dinyatakan sebagai jumlah dari n bilangan dalam rentang 1 sampai m. Hal berikut bekerja secara rekursif dengan cara yang jelas.

```
>function map countways (k; n, m) ...
```

```
    if n==1 then return k>=1 && k<=m
    else
      sum=0;
      loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
      return sum;
    end;
  endfunction
```

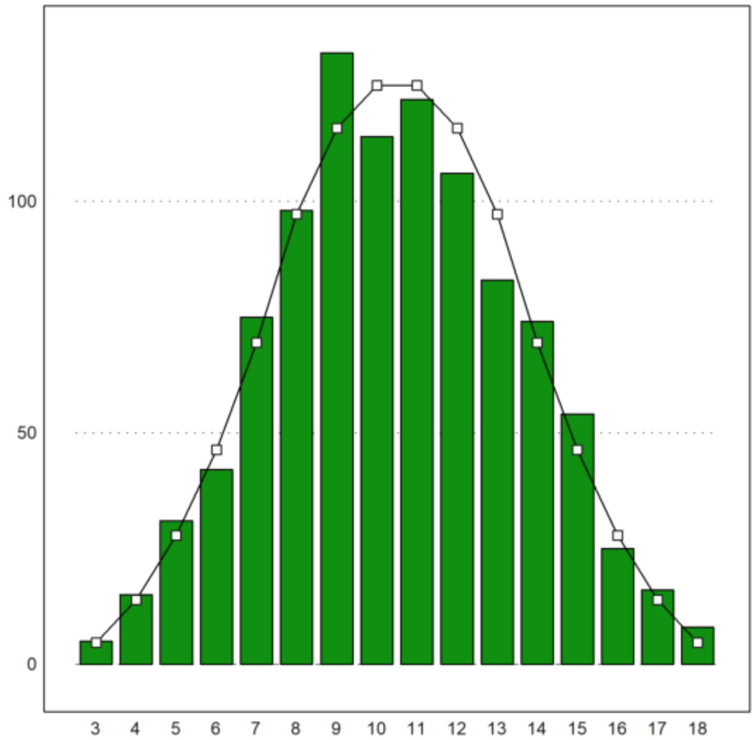
```
>cw=countways(3:18,3,6)
```

```
 [1,  3,  6,  10,  15,  21,  25,  27,  27,  25,  21,  15,  10,  6,  3,
 1]
```

Nilai yang diharapkan dapat direpresentasikan ke dalam plot.
Fungsi countways menghitung jumlah cara di mana bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 hingga m.

Fungsi ini bekerja secara rekursif dengan cara yang jelas. Jika n=1, maka fungsi mengembalikan k>=1 && k<=m, jika tidak, fungsi akan menghitung jumlah cara dengan melakukan iterasi dari 1 hingga m. Hasilnya adalah jumlah cara di mana bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 hingga m.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Uji Chi Square

Untuk melakukan uji chiˆ2 (chi-squared) kita menggunakan fungsi chitest(). Fungsi ini digunakan untuk membandingkan frekuensi observasi dengan frekuensi yang diharapkan. Dalam konteks ini, vektor x merepresentasikan frekuensi observasi, sedangkan vektor y merepresentasikan frekuensi yang diharapkan. Misalnya, jika dari sampel 100 orang ditemukan 40 pria, maka vektor observasi x adalah [40,60], dan vektor harapan y mungkin [50,50]. Uji chiˆ2 digunakan untuk menilai sejauh mana sampel tersebut sesuai dengan frekuensi yang diharapkan.

Sebagai contoh, kami menguji lemparan dadu untuk distribusi baju. Pada 600 lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

```
0.498830517952
```

Fungsi ini digunakan untuk menguji asosiasi antara dua set data. Set pertama [90,103,114,101,103,89] mewakili jumlah yang diamati, dan set kedua dup(100,6) mewakili jumlah yang diharapkan. Hasil dari fungsi chitest adalah probabilitas yang terkait dengan statistik uji chi-kuadrat, yang menunjukkan kemungkinan bahwa data kategoris yang diamati diambil dari distribusi yang diharapkan

Tes chi-kuadrat juga memiliki mode, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harus hampir sama. Parameter >p menginterpretasikan vektor-y sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

```
0.497
```

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak distribusi baju. Ini tidak membuktikan bahwa dadu kami adil. Tapi kita tidak bisa menolak hipotesis kita.

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan tes yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

```
0.594554930686
```

Fungsi random([1,n*6]) digunakan untuk menghasilkan serangkaian bilangan acak antara 1 dan 6 sebanyak n kali. Kemudian, fungsi count(t*6,6) digunakan untuk menghitung berapa kali angka 6 muncul dalam serangkaian bilangan acak tersebut. Fungsi dup(n,6) digunakan untuk menghasilkan serangkaian bilangan 6 sebanyak n kali. Hasil dari fungsi chitest(count(t*6,6),dup(n,6)) akan mengembalikan nilai p, yang dapat digunakan untuk menentukan apakah serangkaian bilangan acak tersebut terdistribusi secara merata atau tidak.

Uji T

Uji-t adalah sebuah metode statistik yang digunakan untuk menguji perbedaan signifikan antara rata-rata dua kelompok yang berbeda.

Misalnya, akan dilakukan pengujian nilai rata-rata dari 100 elemen data menggunakan uji-t.
Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

```
0.161169176307
```

Fungsi ttest() membutuhkan nilai rata-rata, simpangan, jumlah data, dan nilai rata-rata yang akan diuji. Dalam syntax tersebut, normal([1,100]) menghasilkan 100 angka acak yang diambil dari distribusi normal dengan nilai rata-rata 0 dan simpangan baku 1. Kemudian, nilai-nilai tersebut dikalikan dengan 10 dan ditambahkan dengan 200 untuk menghasilkan sampel data dengan nilai rata-rata 200 dan simpangan baku 10.

Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kami menolak hipotesis bahwa mereka memiliki rata-rata yang sama, jika hasilnya $<$0,05.

```
>tcomparedata(normal(1,10),normal(1,10))
```

```
0.23564235612
```

Jika kita menambahkan bias ke satu distribusi, kita mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

```
0.000114481122387
```

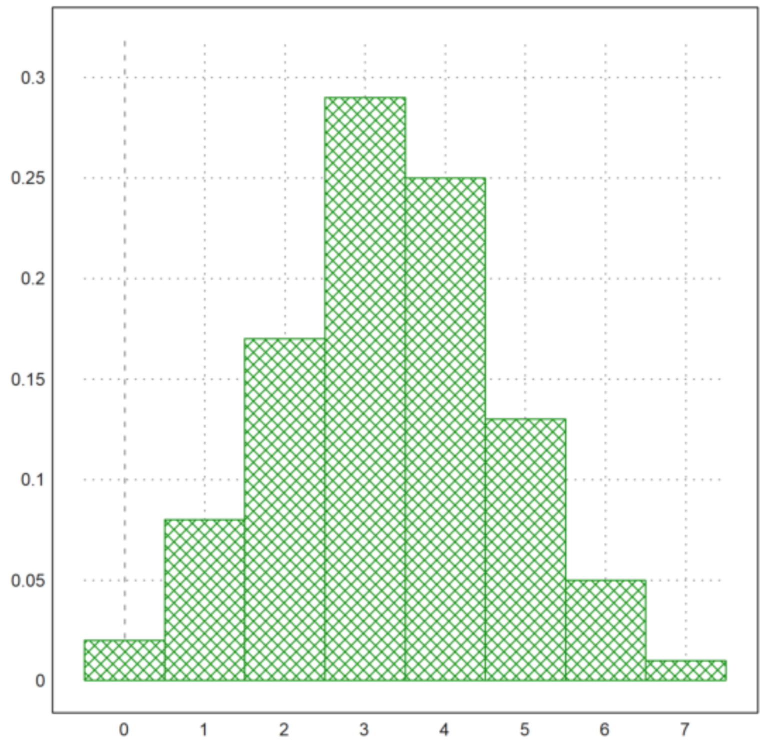
Pada contoh berikutnya, kita menghasilkan 20 lemparan dadu acak sebanyak 100 kali dan menghitung yang ada di dalamnya. Harus ada 20/6=3,3 yang rata-rata.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

```
3.31
```

Bandingkan jumlah satu dengan distribusi binomial. Pertama lakukan plot distribusi satuan.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\/"):
```




```
>t=count(R,21);
```

Menghitung nilai yang diharapkan

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-kuadrat menolak hipotesis bahwa distribusi kami adalah distribusi binomial, jika hasilnya <0,05.

```
>chitest(t1,b1)
```

```
0.45755878393
```

Uji Independensi

Uji independensi adalah salah satu uji statistik yang digunakan untuk mengetahui apakah ada hubungan antara dua variabel kategorik, dengan kata lain untuk mengetahui independensi antara variabel baris dan kolom. Uji ini berguna untuk mengukur perbedaan pengamatan dan menaksir frekuensi suatu pengamatan dalam kategori tertentu.

Contoh berikut berisi hasil dua kelompok orang (laki-laki dan perempuan, katakanlah) memberikan suara untuk 6 jenis minuman kaleng

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76];  ...
>  writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

```
         1     2     3     4     5     6
m       23    37    43    52    64    74
f       27    39    41    49    63    76
```

Tes tabel chi^2 dapat melakukan uji independensi suara dari jenis kelamin

```
>tabletest(A)
```

```
0.990701632326
```

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
>writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

```
         1     2     3     4     5     6
m     24.9  37.9  41.9  50.3  63.3  74.7
f     25.1  38.1  42.1  50.7  63.7  75.3
```

Kita dapat menghitung koefisien kontingensi yang dikoreksi. Karena sangat dekat dengan 0, kami menyimpulkan bahwa pemilihan jenis minuman kaleng tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Metode Anova

Selanjutnya kami menggunakan analisis varians (Uji-F) untuk menguji tiga sampel data yang terdistribusi normal untuk nilai rata-rata yang sama. Metode tersebut disebut ANOVA (analisis varians). Di Euler, fungsi varanalysis() digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1)
```

```
106.545454545
```

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

```
119.111111111
```

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

```
116.3
```

```
>varanalysis(x1,x2,x3)
```

```
0.0138048221371
```

Dalam pengujian hipotesis statistik di atas, hipotesis nol mengasumsikan bahwa nilai rata-rata ke-3 set data (x1,x2,x3) sama. Namun, berdasarkan analisis varians telah kita lakukan, hipotesis nol dapat ditolak dengan probabilitas kesalahan 1,3%. Ini berarti bahwa ada perbedaan yang signifikan secara statistik antara rata-rata dari ketiga set angka.

Nama : Ardan Andhirta
NIM : 22305141045
Kelas : Matematika E

Sub Topik 9 : Menyimpan Data Hasil Analisis

Data-data yang kita gunakan dalam melakukan analisis statistika dapat kita simpan ke dalam suatu file sehingga ketika kelak ingin digunakan lagi, data tersebut masih ada di file penyimpanan kita. Tak hanya itu, hasil dari analisis statistika yang sudah kita lakukan pun dapat kita simpan ke dalam suatu file.

Berikut adalah cara menyimpan/menulis data ke suatu file.

```
>a=random(1,100); mean(a); dev(a);
>filename="Simpan";
```

Seletah memberi nama untuk file, kita akan menulis vektor a ke dalam file dengan menggunakan fungsi writematrix() dan menggunakan fungsi readmatrix() untuk membaca data.

```
>writematrix(a',filename);
>a=readmatrix(filename)';
```

Kita juga bisa menghapus file yang sudah tersimpan dengan menggunakan fileremove

```
>fileremove(filename);
```

Kemudian kita akan mencoba untuk menggantikan data baru ke file lama dengan menghapus semua data lama, dan menulis lagi data baru yang akan disimpan.

```
>file="Simpan"; open(file,"w");
>writeln("A,B,C"); writematrix(random(3,3));
>close();
>printfile(file)
```

```
A,B,C
0.8351051327697636,0.08458248162153686,0.5192250558799737
0.8548977070796793,0.7679427770316303,0.4018472121828296
0.02466356619093713,0.4253015574769268,0.1649367711598173
```

Selain itu kita juga bisa menyimpan dalam bentuk excel

```
>file="test.csv";
>M=random(3,3); writematrix(M,file);
```

Berikut adalah isi dari file ini.

```
>printfile(file)
```

```
0.2641349455581374,0.6713124187949838,0.135806558906826
0.3437161954193733,0.7730785630232085,0.05730568239045469
0.3519819858104311,0.1107098010416762,0.4220892269525604
```

CVS ini dapat dibuka pada sistem bahasa Inggris ke dalam Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Tetapi titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari file dengan readmatrix().

```
>readmatrix(file)
```

```
    0.264135      0.671312      0.135807
    0.343716      0.773079     0.0573057
    0.351982       0.11071      0.422089
```