

Nama : Fiorenza Rafa Talitha Sevandra
NIM : 23030630051
Kelas : Matematika E 2023

1. Pengenalan software aplikasi matematika

a. Hal hal yang dipelajari beserta contohnya

- Daftar software matematika dan kegunaanya
- Beberapa software matematika

MATLAB

MATLAB (singkatan dari "matrix laboratory") adalah aplikasi untuk komputasi numerik multiparadigma dan bahasa pemrograman komersial yang dikembangkan oleh MathWorks. MATLAB memungkinkan manipulasi matriks, menggambar grafik fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan antarmuka dengan program yang ditulis dalam bahasa lain.

SPSS

SPSS digunakan oleh berbagai universitas, institusi, dan perusahaan untuk melakukan analisis data. Berikut beberapa contoh penggunaan SPSS, yaitu:

- Melakukan riset pemasaran (market research).
- Analisis data survey atau kuesioner.
- Populer digunakan untuk penelitian akademik mahasiswa.
- Populer digunakan oleh keperluan pemerintahan seperti lembaga BPS.
- Data mining.
- Membantu untuk pengambilan keputusan suatu perusahaan.
- Penelitian kesehatan masyarakat.
- Mendokumentasikan data.
- Representasi data statistik.
- Memprediksi suatu kejadian time series

Maple

Maple adalah sistem perangkat lunak matematika berbasis komputer, yaitu komputer sistem aljabar dari Waterloo Maple Software (WMS) (Tung, 2003:3). Program yang dikembangkan mencakup tentang penyelesaian matematika untuk mendukung berbagai topik operasi matematika yang meliputi analisis numerik, aljabar simbolik, kalkulus, persamaan diferensial, aljabar linier dan grafik untuk melukiskan suatu peristiwa yang sulit teramati atau bersifat abstrak. Maple bersifat simbolik dan mampu memanipulasi solusi aljabar dengan tampilan berbagai mode plot dan berbagai grafik dua dimensi, tiga dimensi, dan animasi.

Graphmatica

Graphmatica merupakan aplikasi untuk menggambar grafik persamaan yang handal, mudah digunakan, dengan fitur numerik dan kalkulus

Math Script Code

Math ScriptConsole (MSC) memungkinkan untuk melakukan penghitungan matematika menggunakan skrip dan perintah dalam antarmuka multi-jendela berbasis console, dapat menjalankan satu perintah secara interaktif atau menjalankan sekumpulan perintah dari file skrip. MSC adalah pengganti ideal untuk kalkulator windows yang terlalu sederhana untuk ilmuwan dan insinyur. Math ScriptConsole

berada di bawah kekuatan dan fleksibilitas mesin Microsoft VBScript. Ini mencakup fungsi tambahan dan perpustakaan yang dirancang khusus untuk ilmuwan dan insinyur. Terlebih lagi, sangat mudah untuk menambahkan fungsi sendiri dan membuat perpustakaan sendiri.

Mathcad Prime

Mathcad adalah perangkat lunak komputer terutama ditujukan untuk verifikasi, validasi, dokumentasi dan penggunaan ulang perhitungan teknik. Pertama diperkenalkan pada tahun 1986 pada MS-DOS, itu adalah orang pertama yang memperkenalkan mengedit langsung dari notasi matematika mengeset, dikombinasikan dengan perhitungan otomatis nya.

GeoGebra

GeoGebra adalah software matematika dinamis untuk semua jenjang pendidikan yang menyatukan geometri, aljabar, spreadsheet, grafik, statistik, dan kalkulus dalam satu paket yang mudah digunakan. GeoGebra adalah software dengan komunitas jutaan pengguna yang berkembang pesat yang tersebar di hampir setiap negara. GeoGebra telah menjadi penyedia terkemuka perangkat lunak matematika dinamis, yang mendukung pendidikan sains, teknologi, teknik, dan matematika (STEM) serta inovasi dalam pengajaran dan pembelajaran di seluruh dunia.

LaTeX

LaTeX adalah software seperti Word yang digunakan untuk membuat dokumen seperti buku, laporan, tesis dan makalah. Perbedaan utama antara LaTeX dan Word adalah Microsoft Word menyediakan teks yang sudah diformat khusus yang bisa dipilih dari menu-menunya sehingga apa yang dilihat di layar monitor akan sangat menyerupai hasilnya ketika dicetak. Pada LaTeX teks masih berbentuk plaintext, yaitu teks yang belum diformat. Proses formatting teks dilakukan dengan menggunakan bahasa markup.

Euler Maths Toolbox

Euler kemungkinan untuk melakukan komputasi numerik, simbolik atau integer, serta statistik dan optimasi. Ekspresi simbolik dan fungsi yang terintegrasi dengan sintaks Euler dan memungkinkan penggunaan bilangan real dan kompleks, aritmatika interval string dan vektor string, solusi untuk sistem linear dan dukungan untuk matriks. Euler Math Toolbox adalah alat yang sangat serbaguna dan membantu yang sangat berguna untuk menghitung persamaan yang kompleks.

Microsoft Mathematics

Microsoft Mathematics adalah aplikasi yang dibuat untuk membantu pelajar dan mahasiswa dalam belajar matematika. Fitur yang terdapat pada Microsoft Mathematics yaitu kalkulator sains, visualisasi grafik secara 2D dan 3D, komputasi operasi aljabar serta komputasi simbolik untuk fungsi-fungsi dan operasi matematika yang elementer. Microsoft Math adalah program edukasi, dibuat untuk sistem operasi Microsoft Windows, yang membantu pengguna untuk menyelesaikan permasalahan matematika and

sains. Dibangun dan diprakarsai oleh Microsoft, Microsoft Math secara pokok ditargetkan untuk pelajar sebagai alat bantu belajar. Add-on yang menawarkan fitur dari Microsoft Math tersedia untuk Microsoft Word 2007. Add-on ini juga diprakarsai oleh Microsoft dan tersedia gratis untuk pengguna Microsoft Word yang berhasil memvalidasi Microsoft Word mereka melalui Genuine Microsoft Program.

R

Fungsi R diantaranya digunakan untuk riset dan akademis, karena software R sangat cocok untuk riset, baik secara statistik, ekonomi, komputasi numerik, dan pemrograman komputer. Karena didukung oleh banyak tenaga ahli di bidangnya masing-masing, R layak digunakan sebagai perangkat lunak yang dijadikan acuan bagi berbagai kalangan, diantaranya kalangan akademik (dosen, mahasiswa).

GNU Octave

GNU Octave adalah bahasa tingkat tinggi, terutama ditujukan untuk komputasi numerik. Software ini menyediakan antarmuka baris perintah yang nyaman untuk memecahkan masalah linier dan nonlinier secara numerik, dan untuk melakukan eksperimen numerik lainnya menggunakan bahasa yang sebagian besar kompatibel dengan Matlab. Octave juga dapat digunakan sebagai bahasa berorientasi batch.

SageMath

SageMath adalah sebuah aplikasi matematika yang dapat digunakan untuk komputasi tingkat dasar sampai tingkat lanjut, matematika terapan maupun matematika teori. Fitur-fitur yang terdapat di dalam SageMath meliputi beberapa aspek matematika seperti aljabar, kombinatorik, teori grafik, teori bilangan, kalkulus, analisis numerik dan statistik. Tujuan dari pengembangan SageMath yaitu agar dapat menggunakan aplikasi-aplikasi matematika tersebut dalam sebuah lembar kerja secara langsung tanpa harus berpindah-pindah aplikasi. Komputasi numerik dapat dilakukan dengan fungsi-fungsi yang disediakan oleh SageMath atau dengan melalui aplikasi untuk komputasi numerik yang diintegrasikan dalam SageMath. SageMath dapat digunakan secara online maupun dengan cara diinstal pada komputer.

Maxima

Maxima merupakan salah satu software open source yang mempunyai kemampuan untuk pembelajaran matematika topik aljabar, kalkulus, aritmetika, dan grafik. Maxima dikembangkan oleh MACSYMA system, dimana Maxima merupakan salah satu Computer Algebra System (CAS) yang mengkombinasikan kemampuan grafis, simbol, dan numerik. Maxima dapat digunakan untuk menyelesaikan pekerjaan-pekerjaan yang berkaitan dengan turunan, integral, persamaan linier, persamaan polynomial, fungsi Laurent, deret Taylor, grafik 2D dan 3D, dan beberapa pekerjaan lainnya

GAP

Software GAP merupakan suatu software yang memuat fungsi, operasi, dan struktur aljabar. Penggunaan software dalam bidang matematika dapat membuat mahasiswa lebih memahami suatu materi.

GAP sebagai pendukung sistem pembelajaran aljabar abstrak yang dilakukan secara tradisional, yaitu cara pengajaran yang fokus utamanya adalah menggunakan konsep - konsep dasar aljabar abstrak, belajar tentang bagaimana pembuktian teorema?teorema aljabar abstrak dan memahami struktur aljabar abstrak.

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi tentang software matematika.
- Mempelajari software matematika yang sudah dicari.
- Membagi software yang ditemukan berdasarkan kategori (komersil, sharaware, freeware, dan open source).
- Membagi software berdasarkan kegunaan (CAS, analisis numerik, geometri, statistika, pendidikan)
- Mencari informasi sejarah singkat software.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam membagi software berdasarkan kategori dan kegunaan, solusinya mencari informasi melalui youtube dan melihat cara kerja software tersebut.
- Kesulitan menentukan software yang cocok untuk menyelesaikan masalah matematika, solusinya mempelajari modul yang telah dibagikan oleh dosen.

>

2. Pengenalan software Euler Maths Toolbox (EMT)

a. Hal hal yang dipelajari beserta contohnya

- Mempelajari software EMT
- Memasang software EMT
- Mempelajari Panduaan singkat dan pengantar penggunaan EMT

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

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

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

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

Komentar (Teks Uraian)

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

- * Judul
- ** Sub-Judul
- latex: $F(x) = \int_a^x f(t) \, dt$
- mathjax: $\frac{x^2-1}{x-1} = x + 1$
- maxima: 'integrate(x^3,x) = integrate(x^3,x) + C
- <http://www.euler-math-toolbox.de>
- See: <http://www.google.de> | Google
- image: logo.png
- ---

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

Judul

Sub-Judul

$$F(x) = \int_a^x f(t) dt$$
$$\frac{x^2 - 1}{x - 1} = x + 1$$
$$\int x^3 dx = \frac{x^4}{4} + C$$

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

Baris Perintah

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

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

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

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

```
4.90873852123
```

```
7.85398163397
```

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

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

1609.344

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

```
kilometer$:=1000;  
km$:=kilometer$;  
cm$:=0.01;  
mm$:=0.001;  
minute$:=60;  
min$:=minute$;  
minutes$:=minute$;  
hour$:=60*minute$;  
h$:=hour$;  
hours$:=hour$;  
day$:=24*hour$;  
days$:=day$;  
d$:=day$;  
year$:=365.2425*day$;  
years$:=year$;  
y$:=year$;  
inch$:=0.0254;  
in$:=inch$;
```

```
feet$:=12*inch$;  
foot$:=feet$;  
ft$:=feet$;  
yard$:=3*feet$;  
yards$:=yard$;  
yd$:=yard$;  
mile$:=1760*yard$;  
miles$:=mile$;  
kg$:=1;  
sec$:=1;  
ha$:=10000;  
Ar$:=100;  
Tagwerk$:=3408;  
Acre$:=4046.8564224;  
pt$:=0.376mm;
```

Untuk konversi ke dan antar unit, EMT menggunakan operator khusus, yakni ->.

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

```
2.48548476895  
101.6 mm
```

Format Tampilan Nilai

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

```
>pi
```

```
3.14159265359
```

```
>longest pi
```

```
3.141592653589793
```

```
>long pi
```

```
3.14159265359
```

```
>short pi
```

3.1416

```
>shortest pi
```

3.1

```
>fraction pi
```

312689/99532

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

1612.7
2.71828182846
3.141592653589793

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

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

```
>longestformat; pi, defformat; pi
```

3.141592653589793
3.14159265359

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

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

0.392857142857
11/28

Perintah Multibaris

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

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

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

```
-0.138444501319
```

```
-3.61155549868
```

Menampilkan Daftar Variabe

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

```
>listvar
```

```
z          2+3i
p1         (x^3+1)/(x+1)
A          Type: Real Matrix (3x3)
r          1.25
a          4
b          15
D          1.73655549868123
v          Type: Real Matrix (1x6)
c          2
w          Type: Real Matrix (1x11)
```

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

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

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

```
>listvar m$
```


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

Menampilkan Panduan

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

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

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

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

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

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

```
>random(10)
```

```
[0.52143, 0.428893, 0.168134, 0.182742, 0.288048, 0.750042,  
0.472935, 0.324407, 0.340388, 0.195494]
```

```
>normal(10)
```

[-0.277172, -1.4568, 0.658046, 1.53204, -2.47296, -0.937381,
0.836566, -0.233827, 0.221796, -0.541047]

Matriks dan Vektor

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

Vektor dan matriks dapat diberi nama seperti variabel biasa.

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

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

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

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

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

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

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

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

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

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

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

```
0.35
```

Bilangan Kompleks

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

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

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

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

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

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

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

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

```
>deg(arg(I)) // 90°
```

```
90
```

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

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

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

```
0+1i
```

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

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

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

```
-1+0i
```

```
0+1i
```


Matematika Simbolik

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

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

$$(b + a)^2$$

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

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

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

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

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

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

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

$$a - b$$

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

3628800

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

3628800

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

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

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

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

```
3 3  
2 5
```

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

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

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

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

10

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

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

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

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

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

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

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

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

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

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

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

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

$$x^2 - x + 1$$

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

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

$$7$$

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

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

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

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

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

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

$$\frac{2x^3 - 3x^2 + 6x}{6}$$

Tampilan Matematika Simbolik dengan LaTeX

Menampilkan hasil perhitunagn simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukan hal ini, tambahkan tanda dolar (\$) di depan tanda & pada setiap perintah Maxima. Perhatikan, hal ini hanya dapat menghasilkan tampilan yang diinginkan apabila komputer sudah terpasang software LaTeX.

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

$$(b + a)^2$$

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

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

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

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


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

$$a - b$$

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi tentang EMT
- Memasang Software EMT
- Belajar menjalankan perintah perintah di EMT.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam memasang software EMT, solusinya dengan melihat panduan cara memasang EMT dan melihat youtube.
- Kesulitan dalam menggunakan beberapa perintah di software EMT, solusinya dengan mempelajari materi yang ada di besmart dan melihat panduan EMT.

>

3. Penggunaan software EMT untuk aplikasi Aljabar

a. Hal hal yang dipelajari beserta contohnya

- Melakukan operasi bentuk-bentuk aljabar
- Melakukan perhitungan dengan berbagai operasi dan fungsi matematika
- Melakukan perhitungan menggunakan bilangan kompleks
- Melakukan perhitungan menggunakan fungsi-fungsi buatan sendiri
- Menyelesaikan persamaan dan sistem persamaan
- Menyelesaikan pertidaksamaan dan sistem pertidaksamaan
- Melakukan manipulasi dan perhitungan menggunakan matriks dan vektor
- Menggunakan aljabar untuk menyelesaikan masalah sehari-hari atau dalam matematika dan bidang lain.

Contoh pertama

Menyederhanakan bentuk aljabar:

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

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

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

Menyederhanakan fungsi :

$$2y^2 + 2x^2 - 3y^2 + 2x^2$$

```
>$&2*y^2+2*x^2-3*y^2+2*x^2
```

$$4x^2 - y^2$$

Baris Perintah

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

```
>r:=8; h:=4; pi*r^2*h/3
```

268.082573106

Perintah harus dipisahkan dengan spasi. Baris perintah berikut mencetak kedua hasilnya.

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

```
201.06192983  
402.123859659
```

Baris perintah dieksekusi dalam urutan yang pengguna tekan tombol "return". Jadi, Anda akan mendapatkan nilai baru setiap kali Anda menjalankan baris kedua.

```
>x :=9;  
>x := cos(x) // nilai cosinus (x dalam radian)
```

```
-0.911130261885
```

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

```
0.612853011666
```

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
```

```
>
```

Struktur kondisional juga berfungsi.

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

```
Thought so!
```

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

Ketika Anda memindahkan kursor sepanjang baris, pasangan-pasangan tanda kurung atau tanda kurung akan disorot. Perhatikan juga baris status. Setelah tanda kurung buka dari fungsi `sqrt()`, baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol "return".

```
>sqrt(sin(90°)/cos(20°))
```

```
1.03158992457
```

```
>3^-7
```

```
0.000457247370828
```

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

Anda juga dapat melakukan klik ganda pada setiap perintah untuk membuka bantuan untuk perintah tersebut. Cobalah melakukan klik ganda pada perintah "exp" di bawah ini dalam baris perintah.

```
>exp(log(2.9))
```

2.9

```
>sin(90°)
```

1

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

```
>longest 1/3
```

```
0.3333333333333333
```

```
>longest 1/9
```

```
0.1111111111111111
```

```
>printhex(1/89)
```

```
2.E05C0B81702E0*16^-2
```

```
>printdual(1/10)
```

```
1.10011001100110011001100110011001100110011001100110011010*2^-4
```

```
>printhex(1/99)
```



```
2.95FAD40A57EB6*16^-2
```

```
>
```

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 digit dan sejumlah tempat (0 untuk keluaran padat), dan secara optimal satu unit.

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

```
Golden Ratio : 1.61803
```

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

```
>none
```

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

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

```
1234.5
```

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

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

```
affe  
charlie  
bravo
```

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

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

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

I

Nilai Boolean

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

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

0
1

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

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

1

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

Variabel global dapat digunakan dalam evaluasi.

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

```
12.56637061435917
```

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

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

```
-0.919535764538
```

Matematika Simbolik

```
>$&44!
```

```
2658271574788448768043625811014615890319638528000000000
```

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

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

```
2481256778
```

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

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

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

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

```
>::: av:g$ av^3;
```

```
      3  
g
```

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

```
      3  x  
x  E
```

```
      3  x  
x  E
```

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

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

$125 E^5$

$125 e^5$

18551.64488782208

```
>fx(5)
```

18551.6448878

Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "fungsi". Ini bisa berupa fungsi satu baris atau fungsi multibaris.

Fungsi satu baris dapat berupa numerik atau simbolis. Fungsi satu baris numerik didefinisikan oleh ":=".

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

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

```
>f(2)
```

4.472135955

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

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

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

Fungsi dapat diplot. Alih-alih ekspresi, kita hanya perlu memberikan nama fungsi.
Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus diberikan dalam string.

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

0.786151377757

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

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

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

0.707106781187

Lebih baik kita menghapus redefinisi dosa ini.

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

0.707106781187

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 menyimpannya juga. Ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

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

16

Untuk meringkas

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

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

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

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

```
1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

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

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

Menyelesaikan Pertidaksamaan

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

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

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

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

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

```
>A=[1,2;80,9]
```

1	2
80	9

Produk matriks dilambangkan dengan titik.

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

3
4

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

[3, 4]

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

```
-0.0596026    0.013245  
0.529801    -0.00662252
```

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

```
11  
276
```

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

```
1    0  
0    1
```

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

```
>A.A
```

```
161    20  
800    241
```



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

1	4
6400	81

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

1761	502
20080	3769

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

Menyortir dan Mengacak

Fungsi `sort()` mengurutkan vektor baris.

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

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

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

Mari kita mengocok vektor.

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

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

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 $A \backslash b$ 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))
```

```
2.666755705149626e-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

Matriks Simbolik

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
```

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

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

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

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

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

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

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

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

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

$$\left[x = \frac{3 - \sqrt{4ab+1}}{2}, x = \frac{\sqrt{4ab+1}+3}{2} \right]$$

$$\left[x = \frac{3 - \sqrt{4ab+1}}{2}, x = \frac{\sqrt{4ab+1}+3}{2} \right]$$

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

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

$$[[a-1, a+1], [1, 1]]$$

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

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

$$[[[a - 1, a + 1], [1, 1]], [[1, -1], [1, 1]]]$$
$$[1, -1]$$

Penggunaan dalam kehidupan sehari-hari

Demo - Suku Bunga

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
```

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari materi diinternet mengenai aljabar dan penggunaan dalam sehari-hari
- Mencari perintah EMT aljabar di website
- Membuat latihan soal berdasarkan soal yang ditemukan di buku.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam memahami perintah di EMT, solusinya dengan mencari informasi di internet.

>

4. Penggunaan software EMT untuk menggambar grafik 2 dimensi (2D)

- a. Hal hal yang dipelajari beserta contohnya
- Menggambar Grafik Fungsi Satu Variabel dalam Bentuk Ekspresi Langsung,
 - Menggambar Grafik Fungsi Satu Variabel yang Rumusnya Disimpan dalam Variabel Ekspresi,
 - Menggambar Grafik Fungsi Satu Variabel yang Fungsinya Didefinisikan sebagai Fungsi Numerik,
 - Menggambar Grafik Fungsi Satu Variabel yang Fungsinya Didefinisikan sebagai Fungsi Simbolik,
 - Menggambar Beberapa Kurva Sekaligus,
 - Menggambar Beberapa Kurva dalam Satu Bidang Koordinat,
 - Menuliskan Label Sumbu Koordinat, Label Kurva, dan Keterangan Kurva (Legend),
 - Mengatur Ukuran Gambar, Format (Style), dan Warna Kurva,
 - Menggambar Sekumpulan Kurva dengan satu Perintah plot2d,
 - Menggambar Kurva Fungsi Parametrik,
 - Menggambar Kurva Fungsi Implisit,
 - Menggambar Kurva Fungsi Kompleks,
 - Menggambar Daerah yang Dibatasi oleh Beberapa Kurva,
 - Menggambar Segi banyak.

Menggambar Grafik Fungsi Satu Variabel dalam Bentuk Ekspresi

Langsung Ekspresi tunggal

Di dalam program numerik EMT, ekspresi adalah string. Jika ditandai sebagai simbolis, mereka akan mencetak melalui Maxima, jika tidak melalui EMT. Ekspresi dalam string digunakan untuk membuat plot dan banyak fungsi numerik. Untuk ini, variabel dalam ekspresi harus "x".

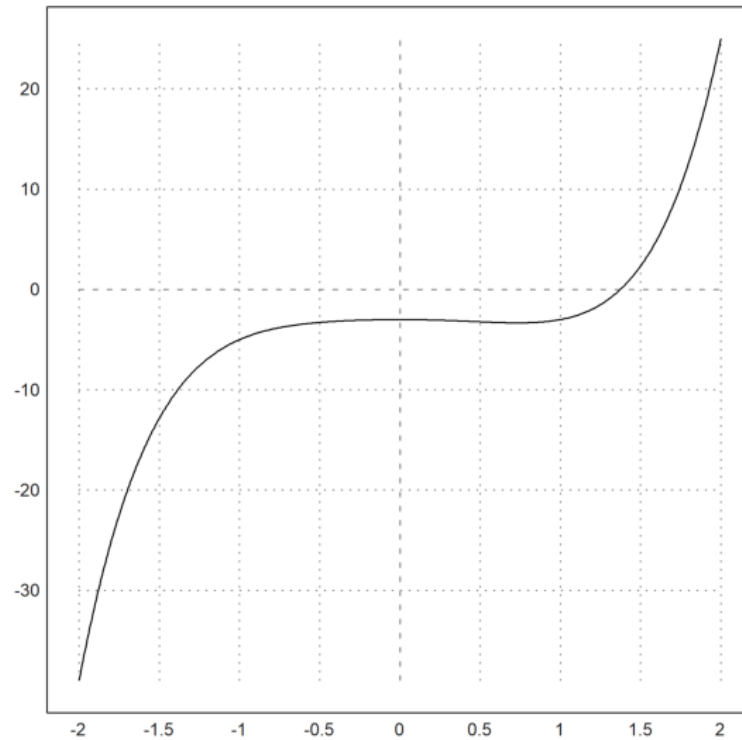
ekspresi dalam string

```
>expr := "x^5-x^2-3"
```

$x^5 - x^2 - 3$

plot ekspresi

```
>plot2d(expr,-2,2) :
```



Menggambar Grafik Fungsi Satu Variabel yang rumusnya Disimpan dalam

Variabel Ekspresi

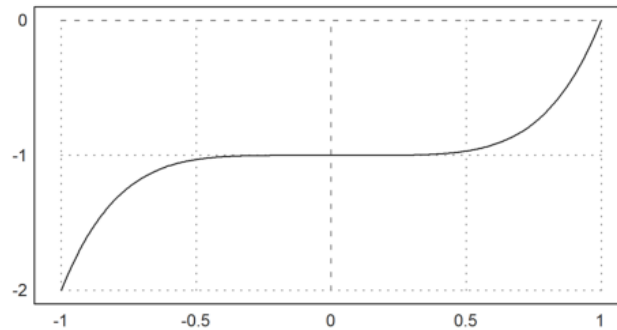
ekspresi

```
>expr &= x^5-1
```

$$x^5 - 1$$

plot dari ekspresi diatas

```
>aspect(2); plot2d(expr,-1,1):
```

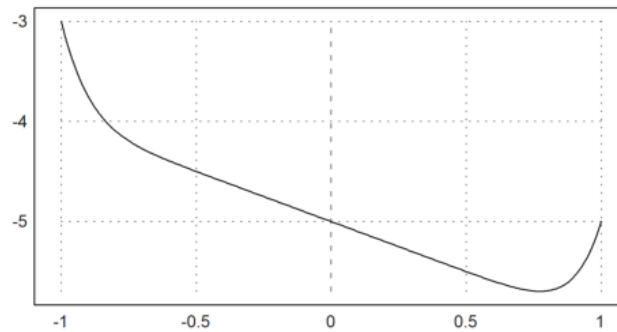


contoh 1

```
>expr := "x^10-x-5"
```

$x^{10}-x-5$

```
>aspect(2) ; plot2d(expr,-1,1):
```



menggunakan variabel lokal

Menggambar Fungsi Numerik

Fungsi Numerik adalah sebuah fungsi dengan himpunan bilangan cacah sebagai domain dan himpunan mendasar yang melibatkan hubungan matematis antara bilangan yang menjadi domain dan bilangan sebagai kodomain.

>

Fungsi numerik memiliki 1 atau lebih variabel independen, yang sering dilambangkan sebagai "X". Variabel X adalah nilai atau parameter yang dapat berubah, dan fungsi numerik menggambarkan bagaimana variabel ini memengaruhi variabel dependen. Variabel dependen adalah hasil perhitungan atau keluaran dari fungsi numerik yang bergantung pada nilai atau perubahan dalam variabel independen.

Dalam EMT cara mendefinisikan fungsi menggunakan syntax function. untuk mendefinisikan fungsi numerik menggunakan tanda " := "

Fungsi numerik menjelaskan bagaimana bilangan dalam domain berhubungan dengan bilangan sebagai kodomain, biasanya diberikan dalam bentuk rumus matematik(persamaan) atau aturan yang memetakan setiap domain kedalam kodomain yang sesuai. contoh:

$$f(x)=2x+3$$

>

$f(x)$ (variabel dependen) adalah fungsi yang memetakan setiap nilai x (variabel independen)kedalam nilai $2x+3$. Terdapat berbagai jenis fungsi yang termasuk ke dalam fungsi numerik, diantaranya:

Fungsi linier dengan bentuk umum

$$f(x) = ax + b$$

Fungsi kuadrat dengan bentuk umum

$$f(x) = ax^2 + bx + c$$

Fungsi eksponensial dengan bentuk umum

$$f(x) = ax$$

Fungsi logaritma dengan bentuk umum

$$f(x) = \log_a(x)$$

Fungsi trigonometri dengan bentuk umum

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

Salah satu cara yang umum digunakan untuk memvisualisasikan fungsi numerik adalah dengan menggambar grafiknya. Grafik ini menggambarkan bagaimana variabel dependen berubah seiring perubahan variabel independen dan membantu dalam memahami sifat-sifat fungsi, seperti titik ekstrim

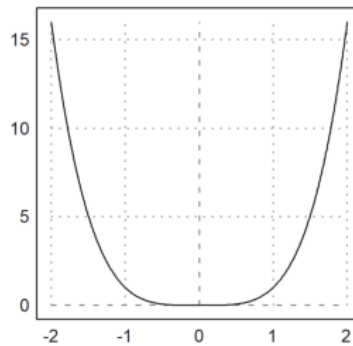
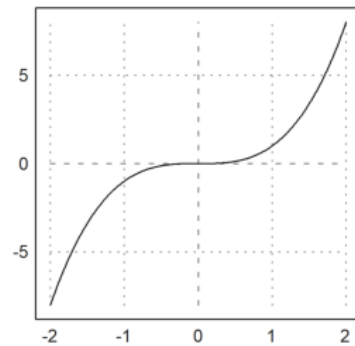
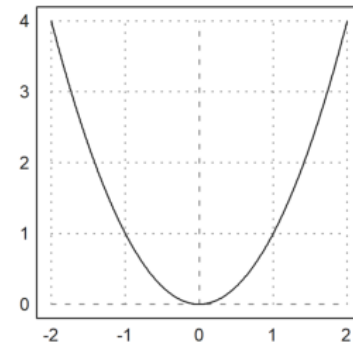
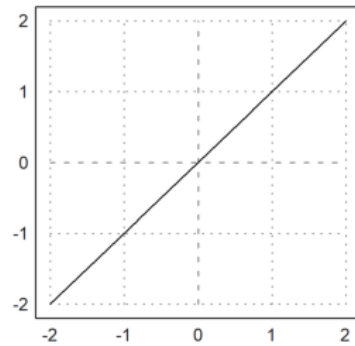
Menggambar Beberapa Kurva Sekaligus

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 bersamaan. 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"+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.

Menggambar Beberapa Kurva pada bidang koordinat yang sama

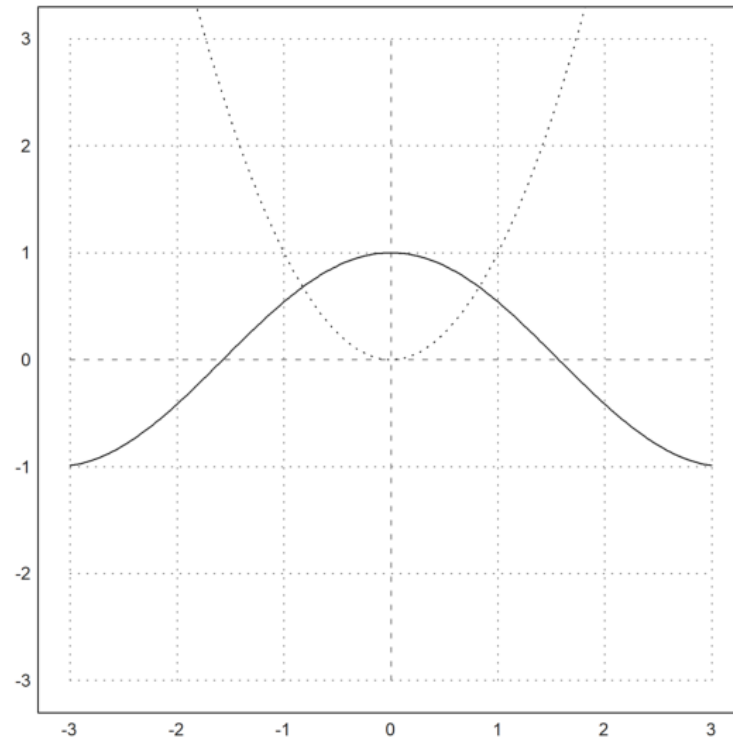
Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah menggunakan `>add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, kecuali panggilan pertama.

Berikut contohnya:
menggambar kurva

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

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

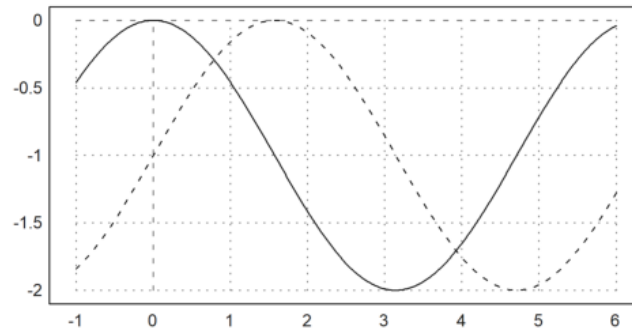
```
>aspect(); plot2d("cos(x)",r=3); plot2d("x^2",style=".",>add):
```

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

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

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



Selain menggunakan `>add` kita juga bisa menambahkannya secara langsung

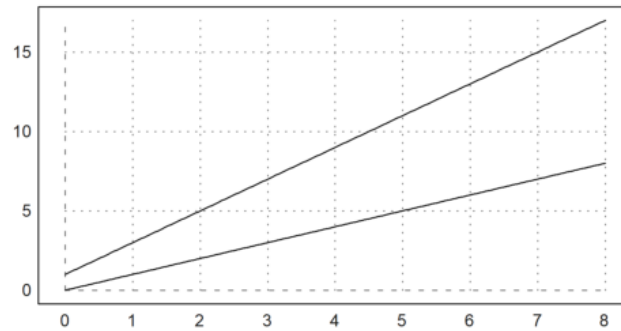
Berikut contohnya:

Menggambar kurva

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

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

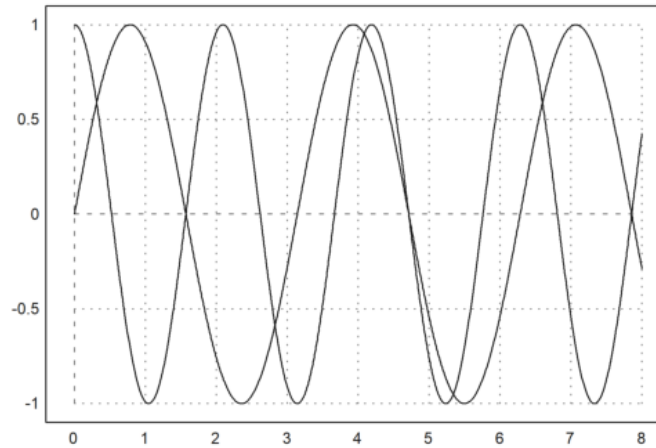
```
>plot2d(["2x+1","x"],0,8):
```



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

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

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



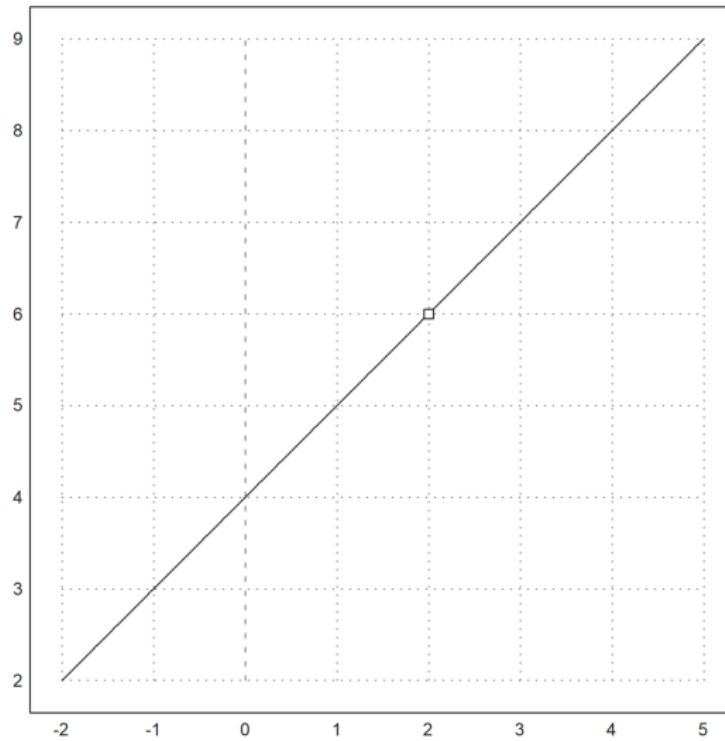
Kegunaan `>add` yang lain juga bisa untuk menambahkan titik pada kurva.

Berikut contohnya:

Menambahkan sebuah titik di

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

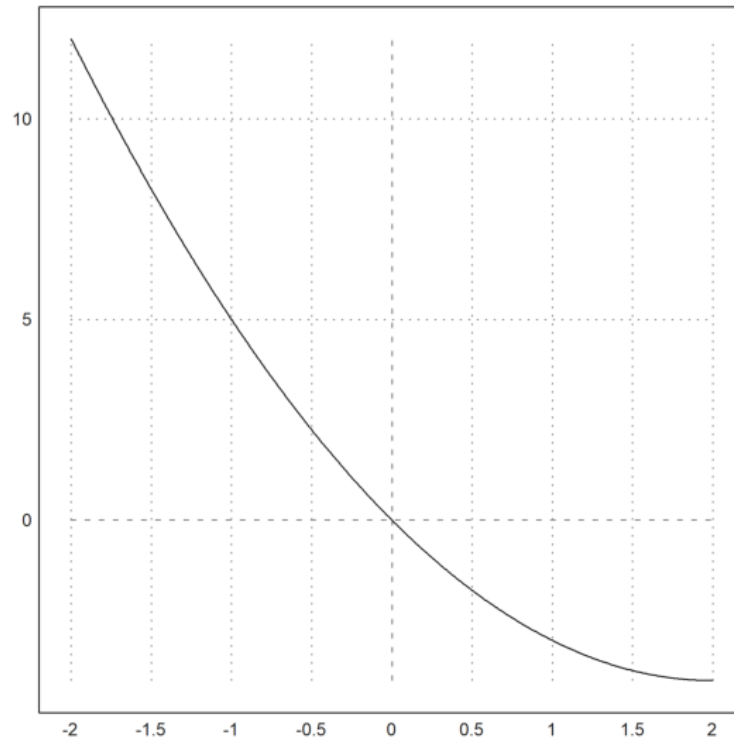
```
>aspect(); plot2d("x+4",-2,5,); plot2d(2,6,>points,>add):
```



Menuliskan Label koordinat, label kurva, dan keterangan

kurva(legend) Dalam EMT, untuk menambahkan judul dapat dilakukan dengan `title="..."`
untuk menambahkan sumbu x dan sumbu y dapat dilakukan dengan `x1="..."`, `y1="..."`
sebagai contoh:

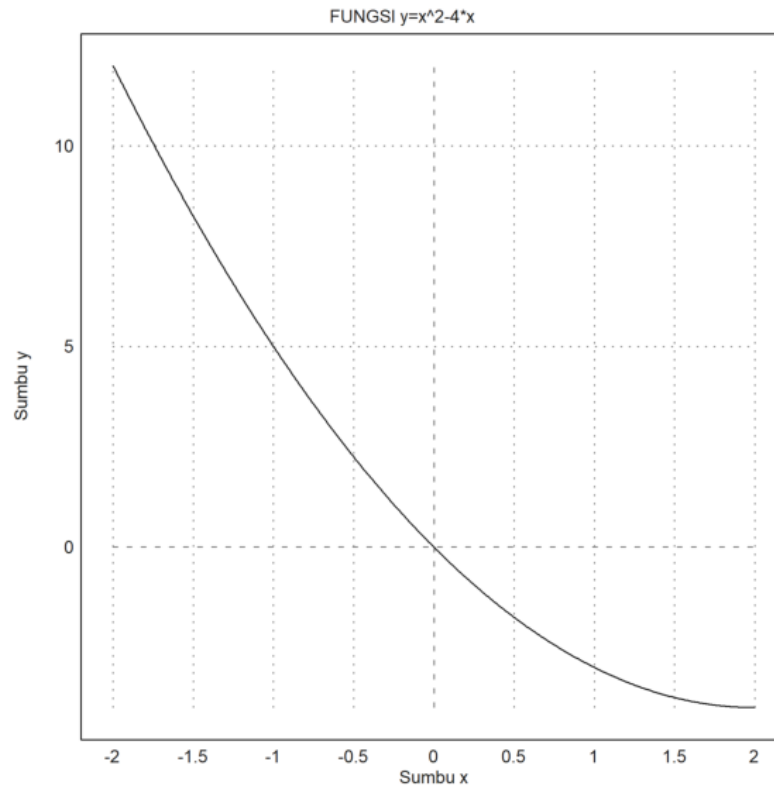
```
>plot2d("x^2-4*x"):
```



untuk menambahkan judul dapat dilakukan dengan title="..."

untuk menambahkan sumbu x dan sumbu y dapat dilakukan dengan x1="...", y1="..."

```
>plot2d("x^2-4*x",title="FUNGSI  $y=x^2-4*x$ ",yl="Sumbu y",xl="Sumbu x"):
```



Mengatur ukuran gambar,format(style),dan warna kurva

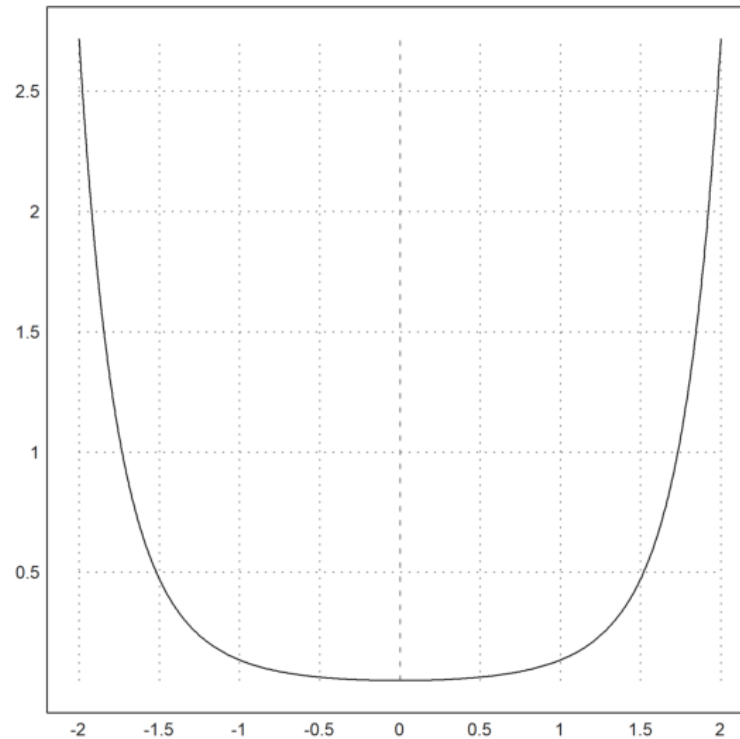
Untuk mengubah ukuran, dapat dilakukan dengan menggunakan `aspect="..."`, semakin besar nilai `aspect`, maka ukuran kurva akan semakin kecil, begitupun sebaliknya

untuk mengganti style, dapat dipilih dengan berbagai pilihan `style="..."`, dapat dipilih dari, misal : `"-"`, `"_"`, `"-."`, `"."`, `"-."`.

untuk warna dapat dipilih sebagai salah satu warna default `color="..."`, warna default= red,green,blue,yellow, dll

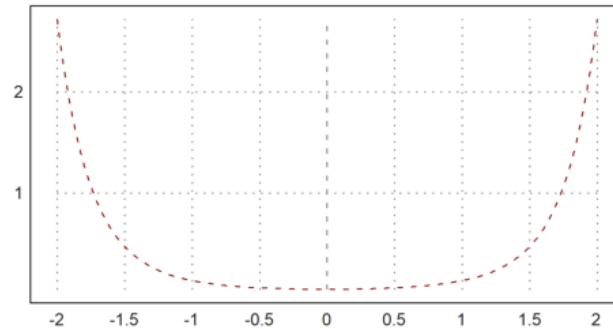
sebagai contoh:

```
>aspect(1); plot2d("exp(x^2-3)");
```

ukuran kurva dapat diganti dengan mengganti nilai `aspect="..."`, semakin besar nilai `aspect`, maka ukuran kurva akan semakin kecil. Untuk mengganti warna dapat ditambahkan dengan `color="..."`, sedangkan untuk mengganti format(`style`) dapat dilakukan dengan menambahkan `style="..."`.

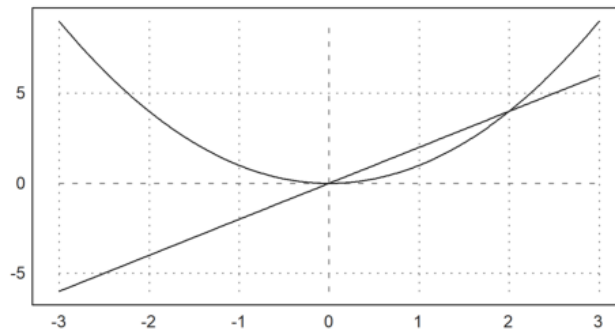
```
>aspect(2); plot2d("exp(x^2-3)", color=red, style="--"):
```



Menggambar Sekumpulan Kurva dalam satu perintah plot2d.

Dalam pembahasan sub-bab 9 kali ini akan membahas mengenai bagaimana menggambar sekumpulan kurva dalam satu perintah plot2d. Menggambar sekumpulan kurva dalam satu perintah plot2d adalah teknik yang digunakan untuk memvisualisasikan beberapa fungsi dalam satu grafik. Ini memudahkan perbandingan antara beberapa kurva.

```
>plot2d(["x^2","2*x"],-3,3):
```



Menggambar Kurva Fungsi Parametrik

kurva yang didefinisikan oleh sebuah persamaan yang menghubungkan koordinat x dan y Contohnya

$$y = x^2$$

Atau

$$x^2 + y^2 = 13$$

dimana persamaan-persamaan ini tidak dikaitkan dengan panjang kurva s , waktu t , dan besaran lainnya. Besaran besaran ini disebut parameter
persamaan parametrik adalah persamaan yang menyatakan hubungan variabel x , y dituliskan dengan

$$x = f(t)$$

$$y = g(t)$$

dengan $a \leq t \leq b$ tiap nilai t menentukan titik (x,y) pada kurva. Jadi , dengan berubahnya nilai t . titik

$$(x, y) = (f(t), g(t))$$

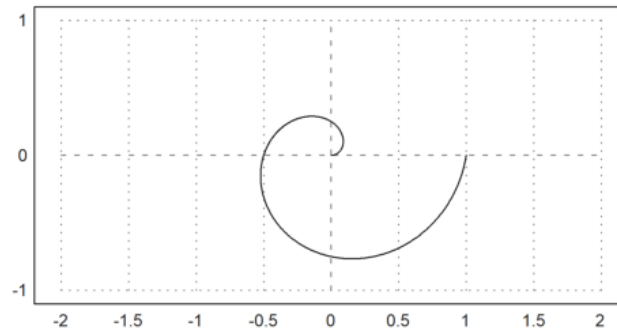
bergerak sepanjang kurva yang disebut kurva parametrik

Dalam contoh berikut, kita memplot spiral

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

Kita perlu menggunakan banyak titik untuk tampilan yang halus

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

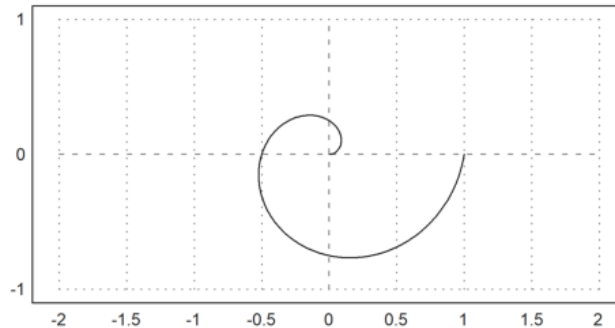


`r` digunakan untuk mengatur radius marker titik-titik yang akan

digunakan dalam plot.

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

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



Menggambar Grafik Bilangan Kompleks

Bilangan kompleks secara visual dapat direpresentasikan sebagai sepasang angka (a, b) membentuk vektor pada diagram yang disebut diagram Argand, mewakili bidang kompleks. Sumbu-x adalah sumbu nyata dan sumbu-y adalah sumbu imajiner.

Menggambar kurva fungsi kompleks sendiri adalah proses visualisasi grafis dari fungsi matematika kompleks (yaitu fungsi yang melibatkan bilangan kompleks, yaitu bilangan dengan bagian real dan imajiner) berperilaku dalam koordinat kompleks. Hal tersebut memungkinkan untuk melihat bagaimana pola, bentuk, dan sifat dari fungsi kompleks tersebut.

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1×2) dalam argumen `cgrid`, hanya garis kisi tersebut yang terlihat.

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

- > Definisi fungsi kompleks, mendefinisikan fungsi kompleks yang dianalisis atau digambarkan. Fungsi ini memiliki variabel kompleks z , yang melibatkan bagian real dan imajiner.
- > Selanjutnya kita dapat menggunakan fungsi `linspace`. Fungsi `linspace` sendiri adalah salah satu fungsi

yang umum digunakan dalam pemrograman, terutama dalam konteks pemrograman numerik dan ilmu data. Ini sering digunakan untuk menghasilkan urutan nilai dalam rentang tertentu dengan jumlah titik yang sama di antara dua titik ujungnya. Penggunaannya tidak terbatas pada pemrosesan sinyal atau elektromagnetik, tetapi bisa digunakan dalam berbagai konteks di mana Anda perlu membuat urutan nilai.

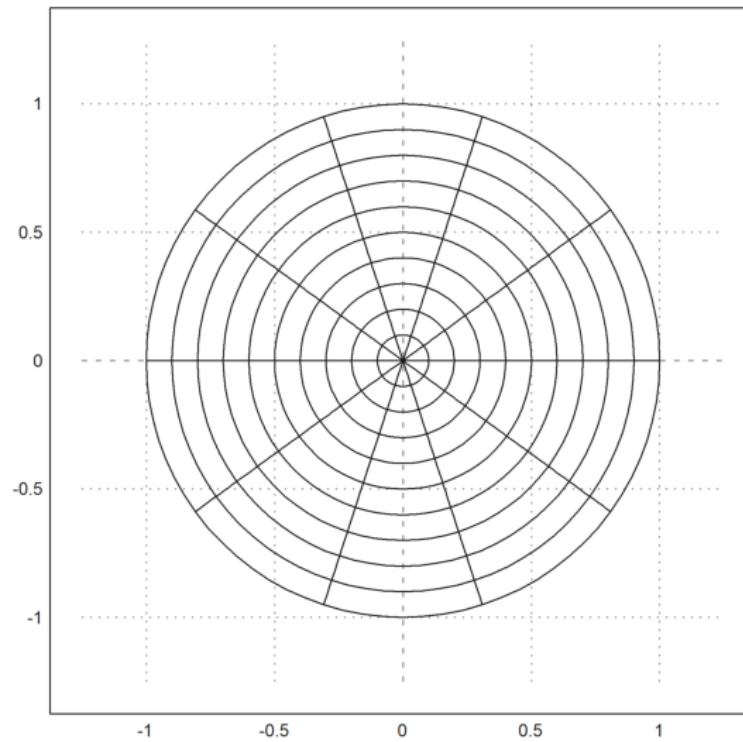
- > Penentuan rentang, memilih rentang nilai z yang ingin ditampilkan di dalam plot. Rentang ini mencakup wilayah kompleks tertentu yang ingin diamati.

- > Menggunakan sintaks `plot2d`.

- > Penyesuaian plot, mengubah plot sesuai yang diinginkan (mengubah warna, format (style), dan sebagainya).

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

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



Penjelasan:

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

Ini adalah perintah untuk membuat plot dari dua fungsi matematika, yaitu $\sin(x)^3$ dan $\cos(x)$, dalam satu plot yang sama. Berikut adalah rincian perintah ini:

" $\sin(x)^3$ " adalah ekspresi pertama yang akan diplotkan. Ini adalah fungsi trigonometri $\sin(x)$ yang dipangkatkan tiga. Fungsi ini tergantung pada variabel x .

" $\cos(x)$ " adalah ekspresi kedua yang akan diplotkan. Ini adalah fungsi trigonometri $\cos(x)$. Fungsi ini juga tergantung pada variabel x .

$xmin=0$ dan $xmax=2\pi$ mengatur rentang (range) plot untuk sumbu x dari 0 hingga 2π . Ini adalah rentang yang akan ditampilkan dalam plot.

$>filled$ mengisi area di bawah kurva fungsi dengan warna, sehingga area di bawah kurva fungsi akan diisi dengan warna.

$style="/"$ mengatur gaya plot menjadi garis miring (" \backslash "). Ini akan menghasilkan plot dengan garis-garis miring.

Menggambar Segi Banyak

Data plot merupakan poligon atau segi banyak. Kita juga dapat membuat kurva atau mengisi kurva. Fungsi perintah yang digunakan untuk menggambar segi banyak atau poligon.

Membentuk poligon dengan fungsi:

```
x=linspace(0,2pi,n); plot2d(cos(x),sin(x),r=1,>filled,style="..."):
```

atau

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

Keterangan

- $filled=true$, mengisi plot.
- $style="..."$: Pilih dari " $...$ ", " $/$ ", " \backslash ", " $\backslash/$ " dan gaya gaya lainnya.
- $fillcolor$: untuk memeberikan warna.

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

Poligon dalam EMT dapat digambar dengan fungsi maksimal. Dengan fungsi maksimal ini, poligon yang dihasilkan dapat berupa poligon tak beraturan.

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

Keterangan:

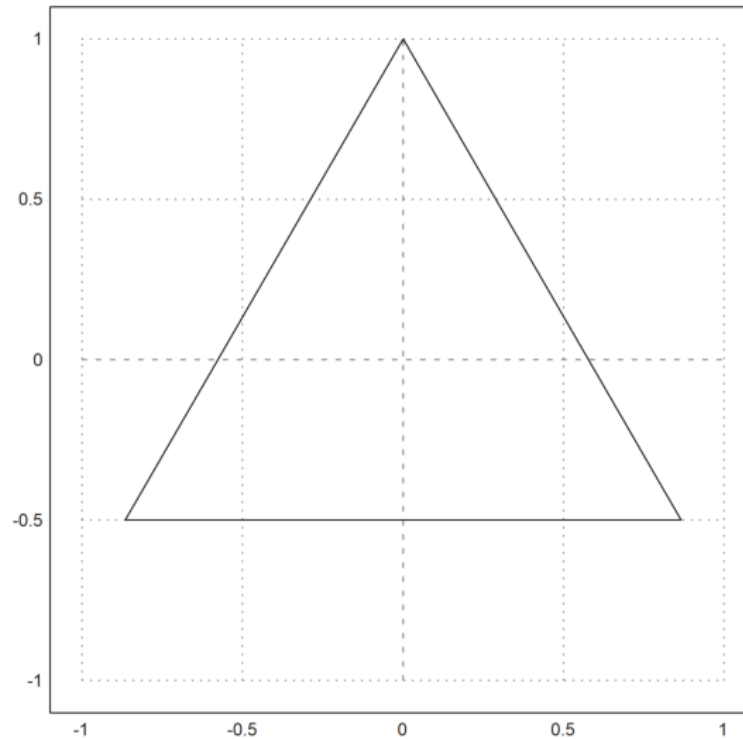
-A adalah titik koordinat dari poligon yang akan dibuat.

-"r" untuk menentukan ukuran bidang koordinat.

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

1. Menggambar Segitiga

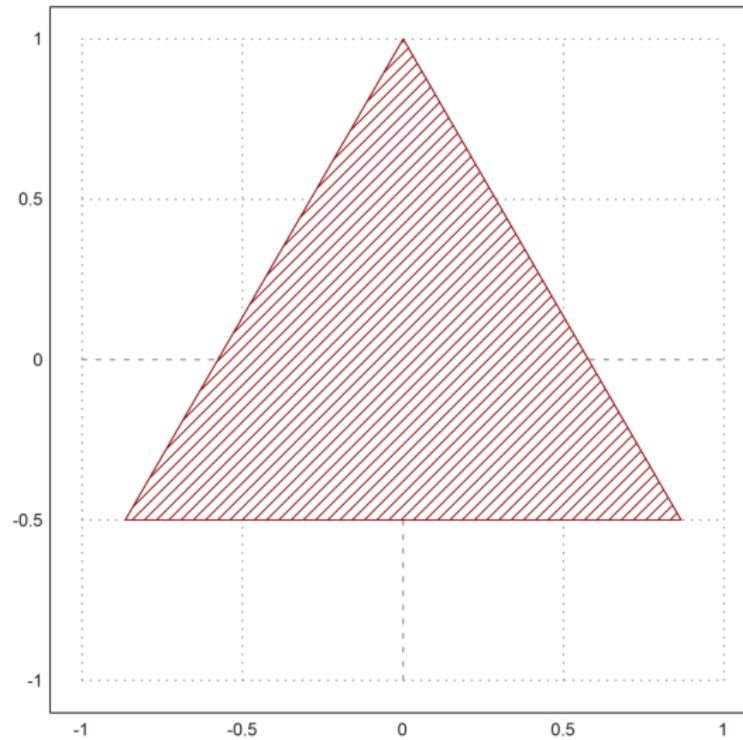
```
>x=linspace(0,2pi,3); ...  
>plot2d(sin(x),cos(x),r=1):
```



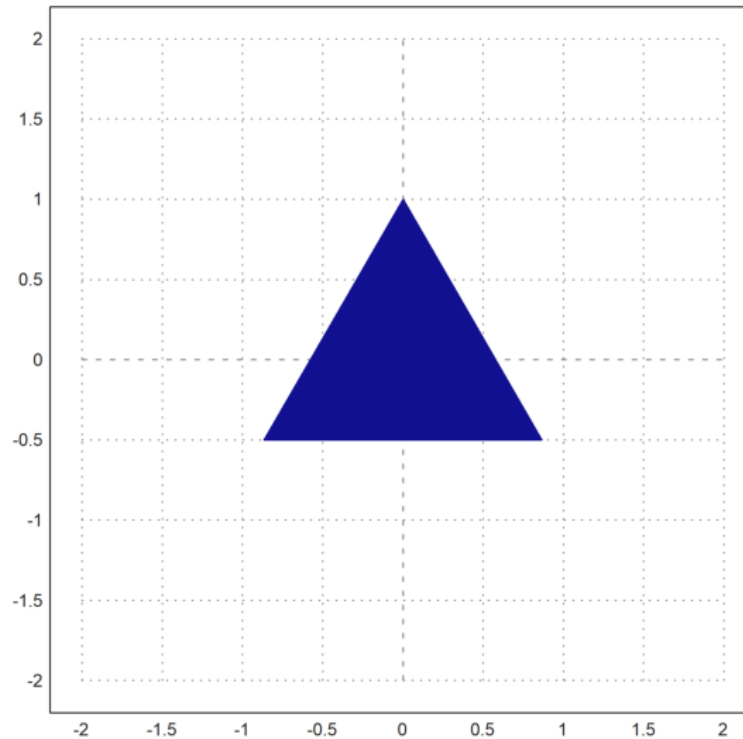
Segitiga diatas digambar dari kurva tertutup dengan 3 titik.

Kita dapat membuat segitiga dengan gaya yang berbeda-beda. Seperti pada contoh berikut ini.

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



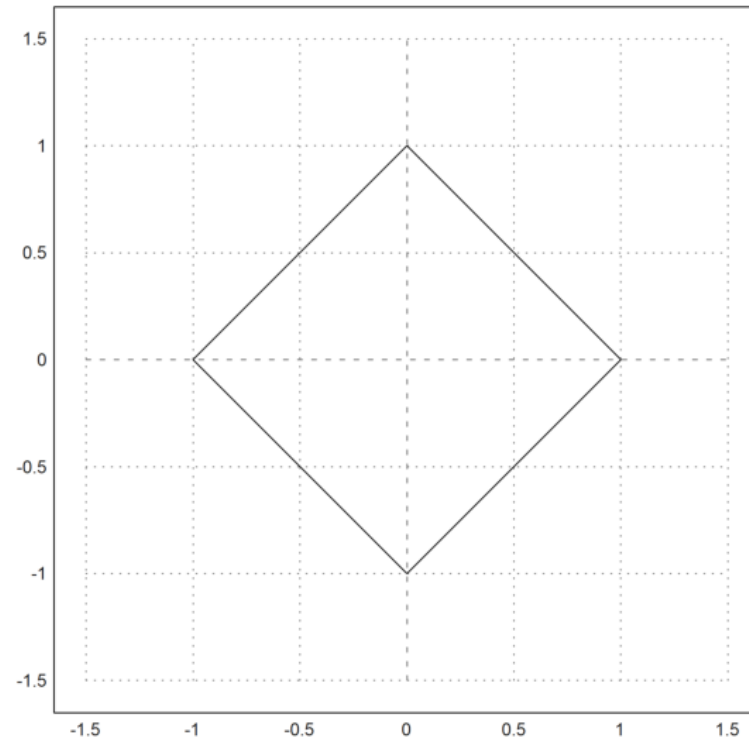
```
>x=linspace(0,2pi,3); ...  
>plot2d(sin(x),cos(x),>filled,style="#",fillcolor=blue,r=2):
```



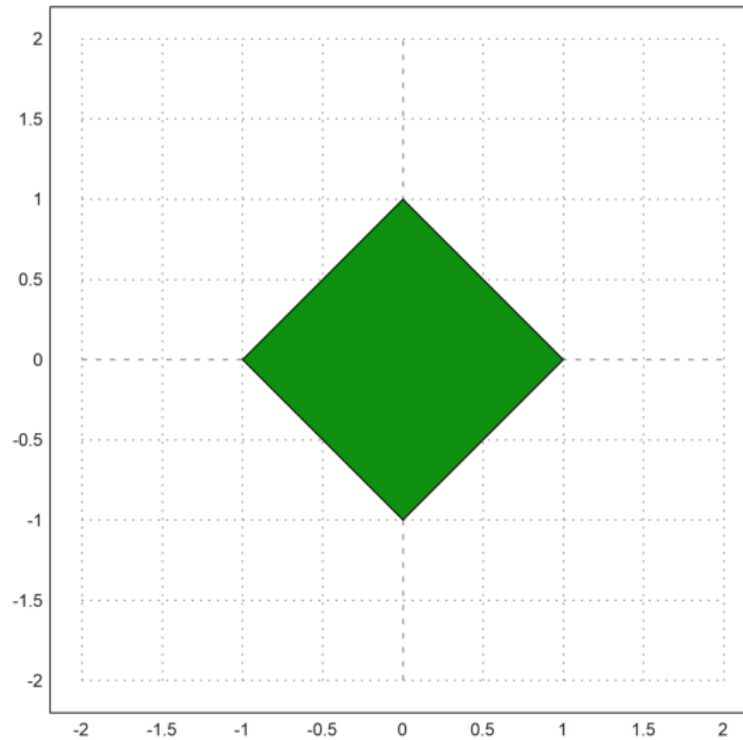
Dua gambar segitiga diatas memiliki gaya yang berbeda, dengan menggunakan fungsi perintah "style=". Gambar segitiga juga dapat dibuat dengan posisi yang berbeda, tergantung pada fungsi yang akan diplot.

2. Menggambar Segiempat

```
>x=linspace(0,2pi,4); ...  
>plot2d(cos(x),sin(x),r=1.5):
```



```
>x=linspace(0,2pi,4);  
>plot2d(cos(x),sin(x),r=2,>filled,outline=1):
```



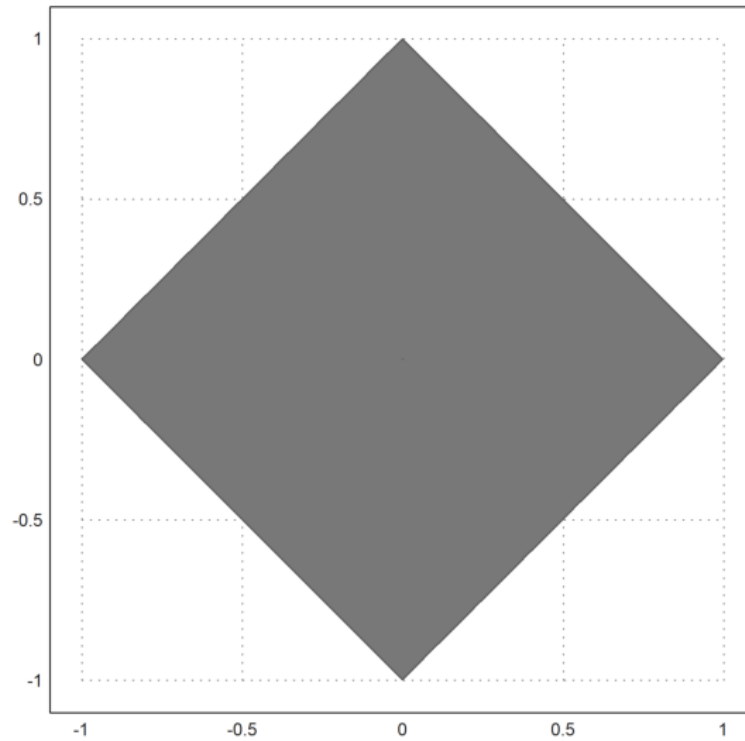
Gambar diatas merupakan salah satu contoh segiempat yang dapat digambar di EMT. Fungsi perintah yang digunakan masih sama seperti fungsi perintah untuk menggambar segitiga.

Selain fungsi perintah diatas, untuk menggambar segi banyak, dapat menggunakan fungsi maksimum.

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



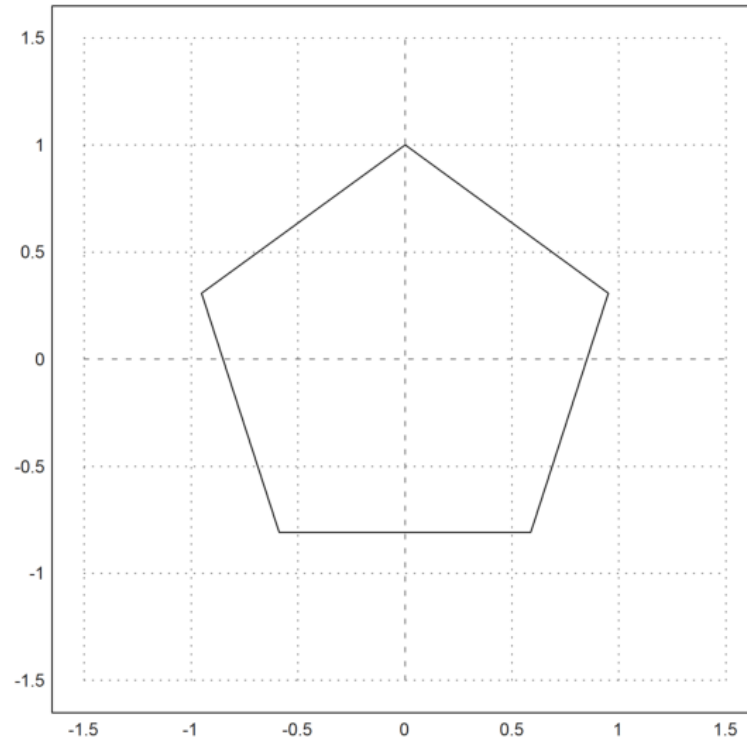

```
>A=[1,1;-1,1;-1,-1;1,-1];  
>function f(x,y) := max([x,y].A');  
>plot2d("f",r=1,level=[0;1],color=gray,n=90):
```



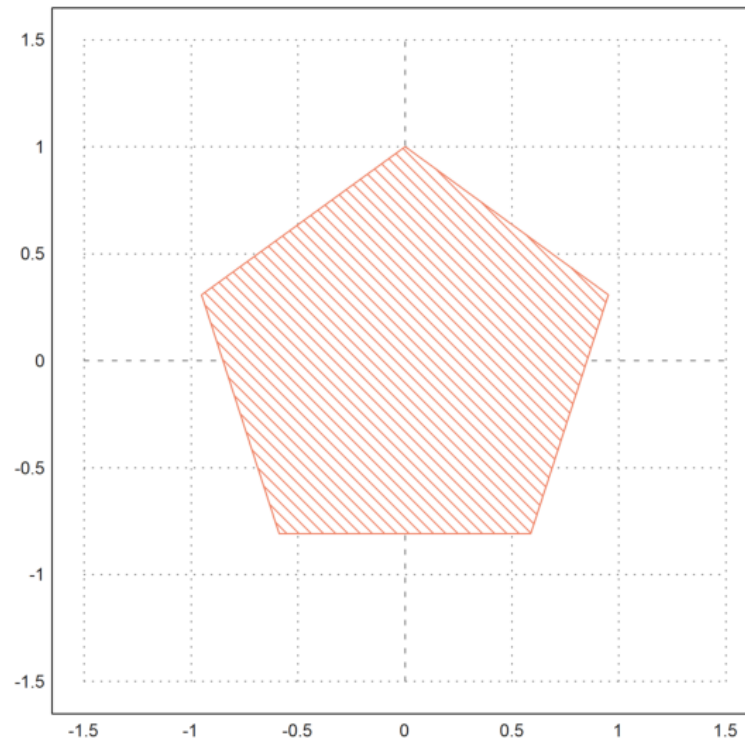
Dengan fungsi maksimal ini, kita dapat menggambar segiempat atau segi banyak sebarang.

3. Menggambar Segilima

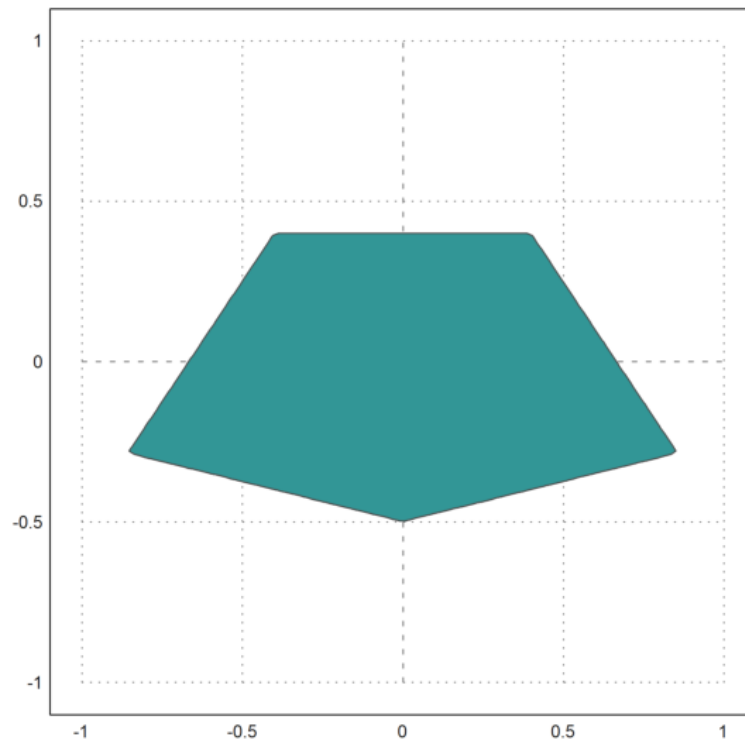
```
>t=linspace(0,2pi,5); plot2d(sin(t),cos(t),r=1.5):
```



```
>t=linspace(0,2pi,5); ...  
>plot2d(sin(t),cos(t),r=1.5,>filled,style="\",fillcolor=orange):
```

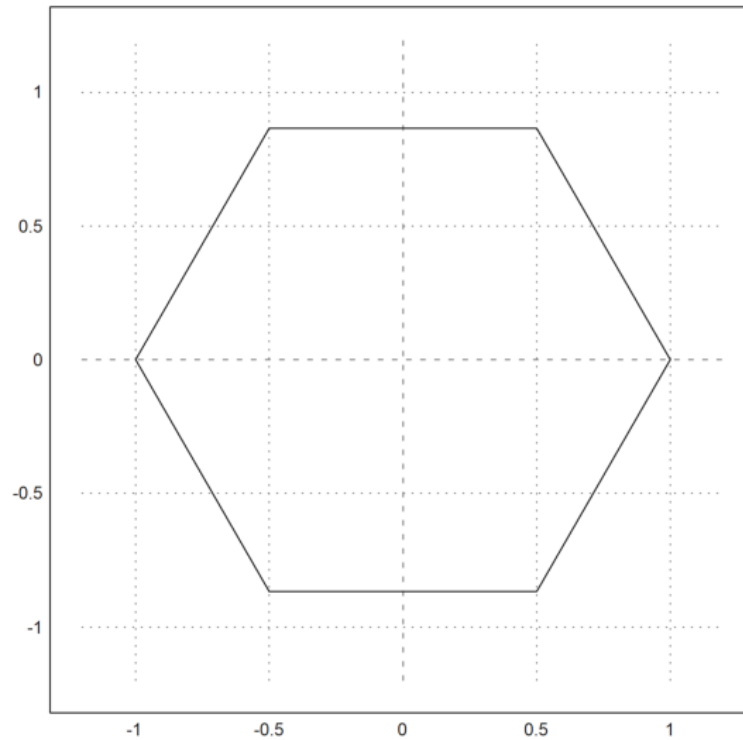


```
>A=[0,5;3,2;1,-4;-1,-4;-3,2];  
>function f(x,y) := max([x,y].A');  
>plot2d("f",r=1,level=[0;2],color=cyan,n=111):
```

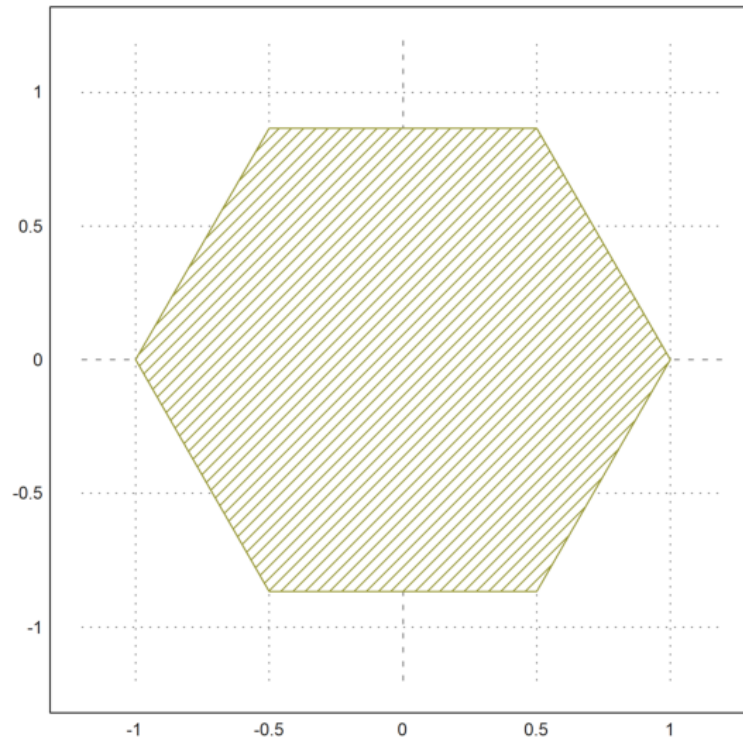


4. Menggambar Segienam

```
>t=linspace(0,2pi,6); ...  
>plot2d(cos(t),sin(t),r=1.2):
```

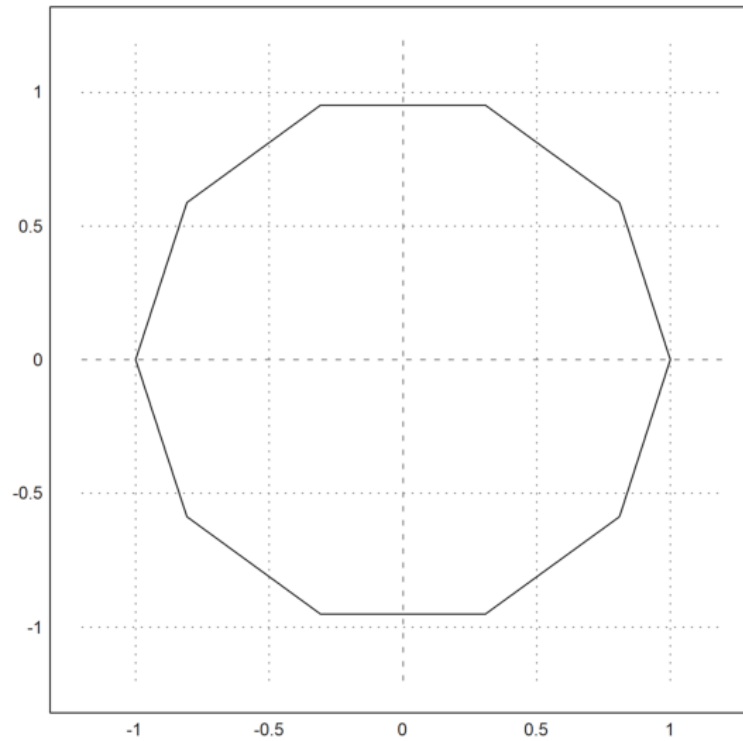


```
>t=linspace(0,2pi,6); ...  
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=olive,r=1.2):
```

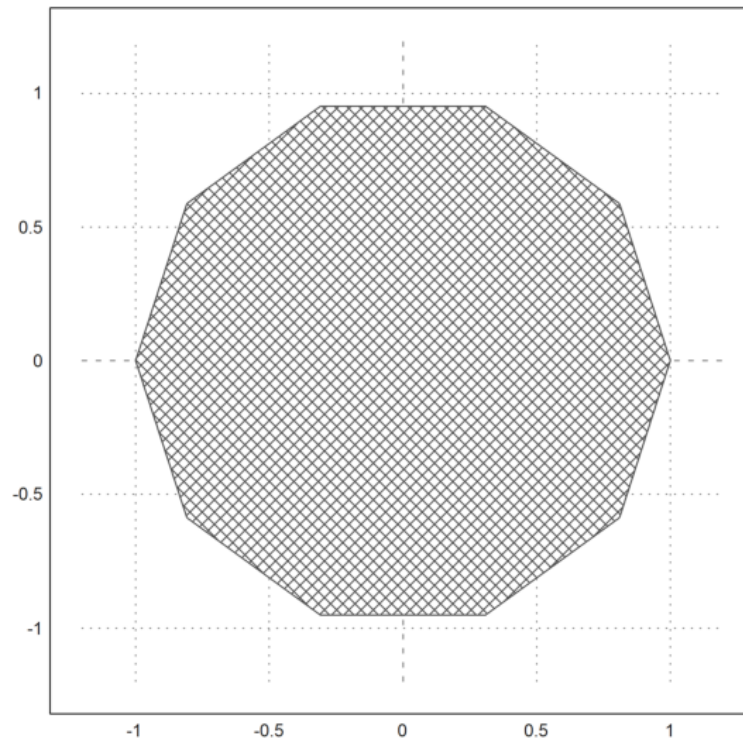


5. Menggambar dekaagon

```
>t=linspace(0,2pi,10); ...  
>plot2d(cos(t),sin(t),r=1.2):
```



```
>t=linspace(0,2pi,10); ...  
>plot2d(cos(t),sin(t),>filled,style="\/",fillcolor=darkgray,r=1.2):
```



- b. Hal hal yang dilakukan dalam mempelajari materi
 - Mempelajari EMT plot2D dan gambar gambar bangun.
 - membuat persamaan yang ditemukan ke aplikasi EMT
- c. Kendala kendala dan usaha untuk mengatasi kendala tersebut
 - Tidak terdapat kendala dalam mempelajari materi ini karena materi mudah dipahami.

5. Penggunaan software EMT untuk menggambar grafik 3

dimensi (3D)

a. Hal hal yang dipelajari beserta contohnya

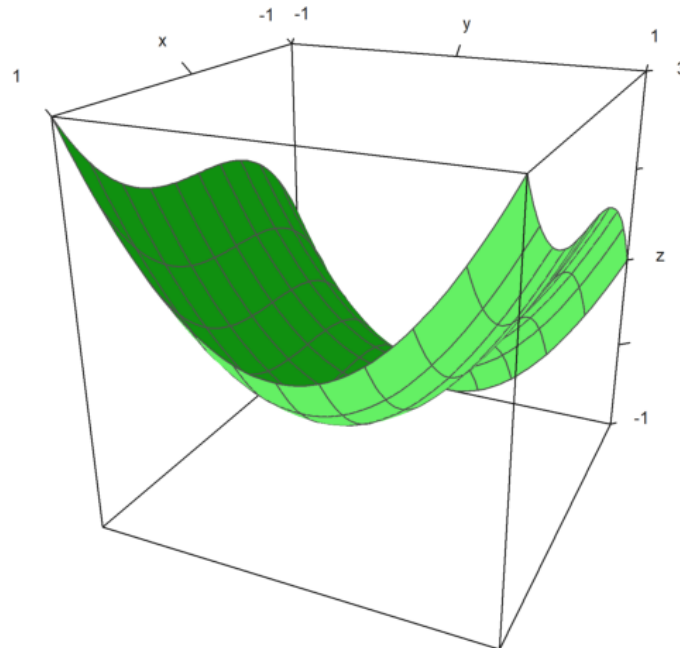
- Menggambar Grafik Fungsi Dua Variabel dalam Bentuk Ekspresi Langsung
- Menggambar Grafik Fungsi Dua Variabel yang Rumusnya Disimpan dalam Variabel Ekspresi
- Menggambar Grafik Fungsi Dua Variabel yang Fungsinya Didefinisikan sebagai Fungsi Numerik
- Menggambar Grafik Fungsi Dua Variabel yang Fungsinya Didefinisikan sebagai Fungsi Simbolik
- Menggambar Data x , y , z pada ruang Tiga Dimensi (3D)
- Membuat Gambar Grafik Tiga Dimensi (3D) yang Bersifat Interaktif dan animasi grafik 3D
- Menggambar Fungsi Parametrik Tiga Dimensi (3D)
- Menggambar Fungsi Implisit Tiga Dimensi (3D)
- Menggambar Titik pada ruang Tiga Dimensi (3D)
- Mengatur tampilan, warna dan sudut pandang gambar permukaan Tiga Dimensi (3D)
- Menggambar Grafik Tiga Dimensi dalam modus anaglif
- Menggambar Diagram Batang Tiga Dimensi
- Menggambar Permukaan Benda Putar

Grafik Fungsi Linear

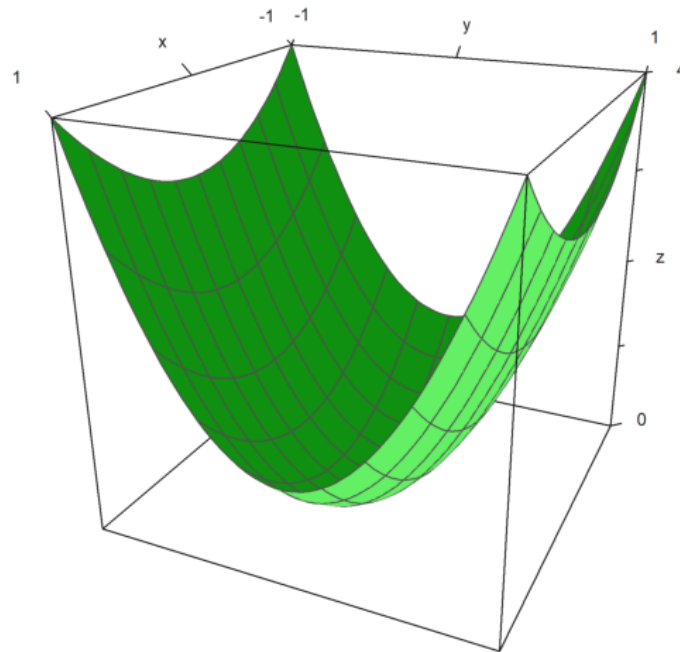
Fungsi linear dua variabel biasanya dinyatakan dalam bentuk

$$f(x, y) = ax + by + c$$

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



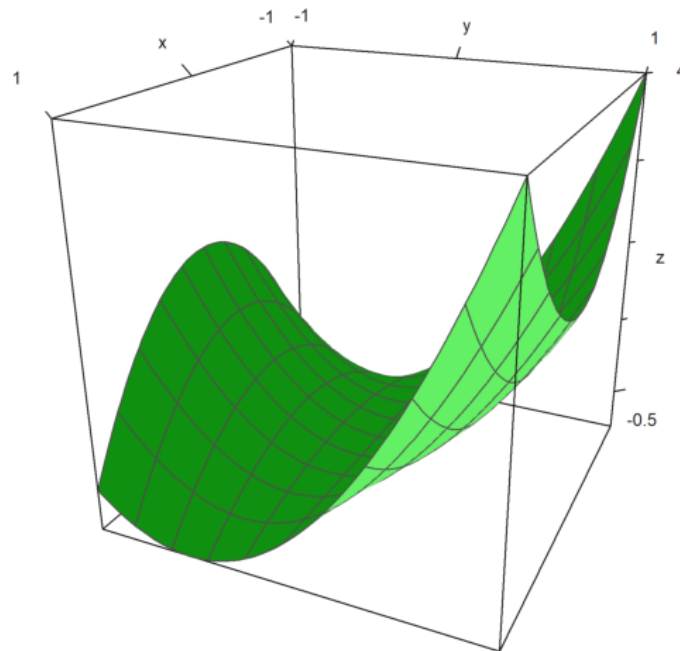
```
>plot3d("x^2+3*y^2"):
```



Fungsi kuadrat dua variabel biasanya dinyatakan dalam bentuk

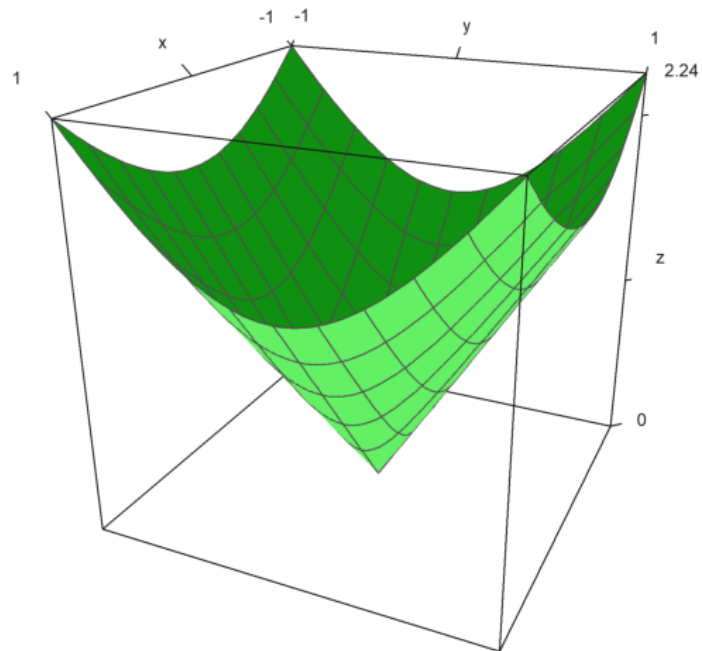
$$f(x, y) = ax^2 + by^2 + cxy + dx + ey + f$$

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



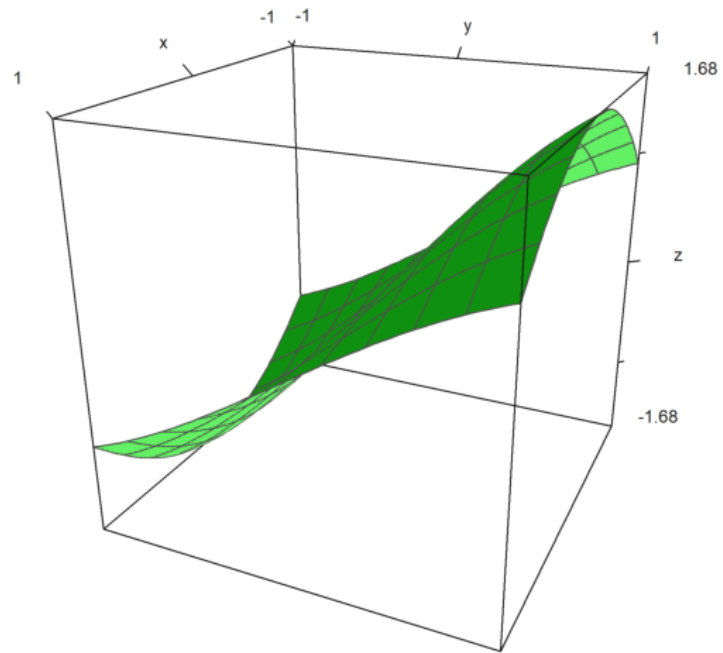
Grafik Fungsi Akar Kuadrat

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

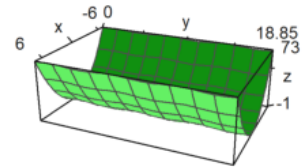


Grafik Fungsi Trigonometri

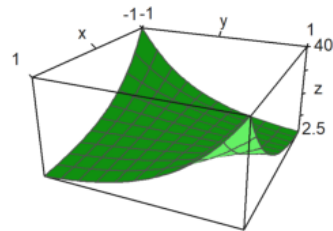
```
>plot3d("2*cos(x)*sin(y)":
```



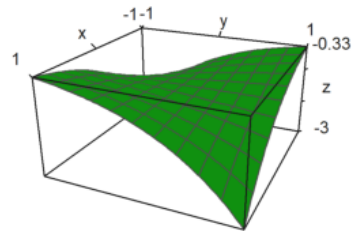
```
>aspect(2); plot3d("2*x^2+sin(y)",-6,6,0,6*pi):
```

```
>plot3d("10*2^(2*x*y)":
```



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

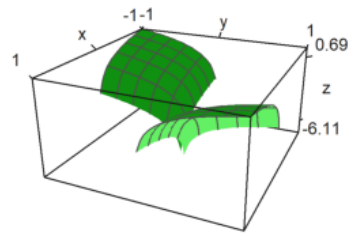


Grafik Fungsi Logaritma

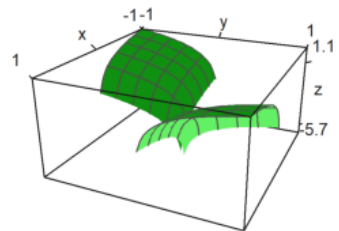
Fungsi logaritma dua variabel bisa dinyatakan sebagai

$$f(x, y) = \log_b(xy)$$

```
>plot3d("log(2*x*y)":
```



```
>plot3d("log(3*x*y)":
```



>

Menggambar Grafik Fungsi Dua Variabel yang disimpan dalam variabel

Ekspresi

contoh:

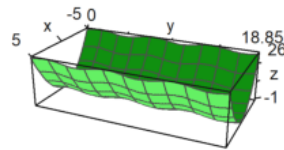
ekspresi dalam string

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

$x^2 + \sin(y)$

plot ekspresi

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



Menggambar Grafik Fungsi Dua Variabel yang * Fungsinya

Didefinisikan sebagai Fungsi Numerik

Fungsi Numerik

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

Contoh:

Fungsi

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

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

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

Gambar Grafik Fungsi

Fungsi satu baris numerik didefinisikan oleh ":=".

Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsi nya didefinisikan sebagai fungsi numerik dalam plot3d:

1. Buat fungsi numerik yang akan digunakan untuk memvisualisasikan data.

```
function f(x,y):=ax+by
```

dimana a dan b adalah konstanta

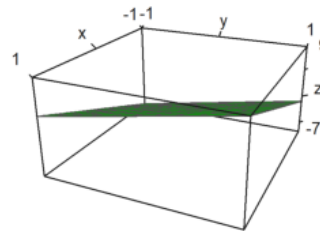
2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi numerik.
plot3d("f"):

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

1. Fungsi Linear Dua Variabel

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

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

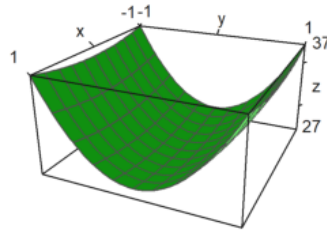


- Fungsi $f(x,y)$ didefinisikan sebagai $5x+3y+1$.
 - Perintah "plot3d("f")" digunakan untuk memplot grafik 3D dari fungsi $f(x,y)$ menggunakan fungsi plot3d di EMT.
 - Grafik yang dihasilkan akan menampilkan fungsi dalam tiga dimensi, dengan sumbu x dan y mewakili variabel masukan dan sumbu z mewakili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

2. Fungsi Kuadrat Dua Variabel

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

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

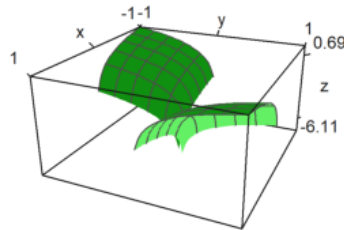


- Perintah "function f(x,y):= x^2+(3*y)^2+27" berarti mendefinisikan fungsi matematika f(x,y) sebagai x pangkat 2 ditambah 3 kali y pangkat 2 ditambah 27.
 - Perintah "plot3d("f")" berarti membuat grafik tiga dimensi dari fungsi f(x,y) yang telah didefinisikan sebelumnya.
-

3. Fungsi Logaritma Dua Variabel

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

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

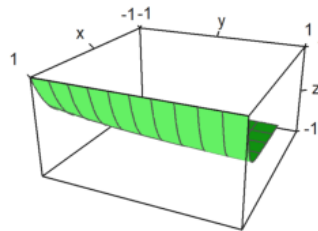


- Input yang diberikan adalah fungsi matematika dua variabel, $f(x,y)$, yang didefinisikan sebagai logaritma hasil kali $2x$ dan y .
 - Perintah "plot3d("f")" digunakan untuk memplot grafik fungsi $f(x,y)$ dalam ruang tiga dimensi.
-

4. Fungsi Eksponen Dua Variabel

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

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

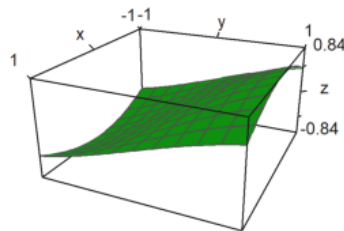


- Perintah 'fungsi $f(x,y) := x^{(5y+10)}$ ' adalah fungsi matematika dua variabel 'x' dan 'y' dan dengan rumus $x^{(5y+10)}$
 - Perintah 'plot3d("f")' digunakan untuk memplot fungsi dalam ruang tiga dimensi. Plot yang dihasilkan akan menampilkan nilai fungsi sebagai permukaan pada bidang x-y, dengan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

5. Fungsi Trigonometri Dua Variabel

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

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



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

Menggambar Grafik Fungsi Dua Variabel yang * Fungsinya

Didefinisikan sebagai Fungsi Simbolik

Fungsi Simbolik

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

Contoh fungsi simbolik yang umum adalah:

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

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

Perbedaan Fungsi Numerik dan Fungsi Simbolik

1. Fungsi Numerik

Fungsi numerik dinyatakan dalam bentuk yang lebih konkret menggunakan angka-angka nyata.

Contoh fungsi numerik adalah

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

dimana kita memberikan nilai numerik kepada "x dan y"

misalnya, $x = 5$ dan $y = 2$, maka hasilnya adalah angka konkret yaitu $g(5,2) = 15$

2. Fungsi Simbolik

Fungsi simbolik dinyatakan menggunakan simbol-simbol matematika seperti huruf (variabel) dan operasi matematika.

Contoh fungsi simbolik adalah

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

"g" adalah simbol fungsi

"x,y" adalah variabel,

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

Gambar Grafik Fungsi

Fungsi satu baris simbolik didefinisikan oleh "=&=".

Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsi nya didefinisikan sebagai fungsi simbolik dalam plot3d:

1. Buat fungsi simbolik yang akan digunakan untuk memvisualisasikan data.

```
function g(x,y)&= ax+by;
```

dimana a dan b adalah konstanta

2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi simbolik

```
plot3d("g"):
```

3. Menentukan rentang variabel

misal

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

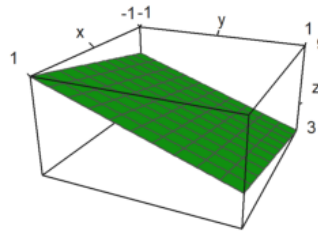
dengan batasan x dari -10 hingga 10 dan batasan y dari -5 hingga 5

Contoh

1. Fungsi Linear Dua Variabel

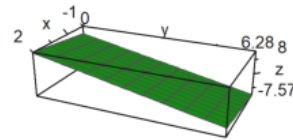
$$g(x,y) = x - 2y + 6$$

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



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

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

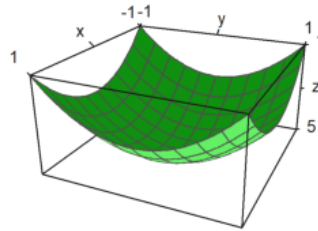


- Perintah "plot3d("g(x,y)",-1,2,0,2*pi)" adalah perintah untuk menggambar grafik fungsi tiga dimensi "g(x,y)" pada rentang x dari -1 hingga 2 dan rentang y dari 0 hingga 2pi.

2. Fungsi Kuadrat Dua Variabel

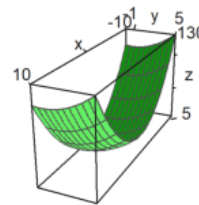
$$g(x, y) = x^2 + y^2 + 5$$

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

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

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

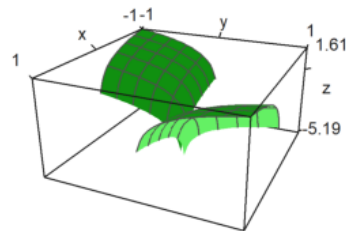


- Perintah "plot3d("g(x,y)",-10,10,-1,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi g(x,y) pada rentang x dari -10 hingga 10 dan rentang y dari -1 hingga 5

3. Fungsi Logaritma Dua Variabel

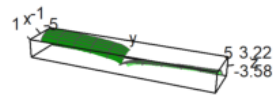
$$g(x,y) = \log(xy5)$$

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



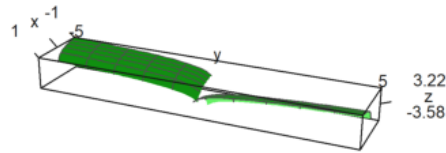
- Fungsi $g(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus logaritma x dikalikan y dikalikan 5
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

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



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

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

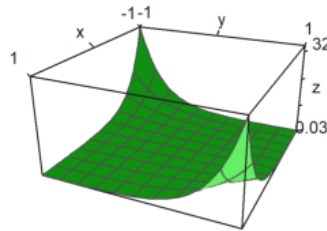


- plot3d: perintah untuk membuat grafik 3D.
 - "g(x,y)": fungsi matematika yang akan digunakan untuk membuat grafik.
 - -1,1: rentang nilai variabel x yang akan digunakan dalam grafik.
 - -5,5: rentang nilai variabel y yang akan digunakan dalam grafik.
 - zoom=4.5: perintah untuk memperbesar tampilan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

4. Fungsi Eksponen Dua Variabel

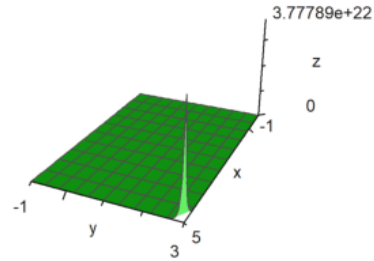
$$g(x, y) = 2^{xy^5}$$

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



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

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

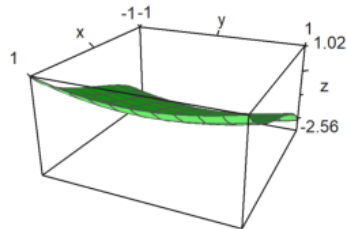


- Perintah `plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3)` adalah perintah untuk membuat plot tiga dimensi dari fungsi 'g(x,y)' dengan batas 'x' dari '-1' hingga '5' dan batas 'y' dari '-1' hingga '3'.
 - `plot3d`: perintah untuk membuat plot tiga dimensi.
 - `"g(x,y)"`: fungsi yang akan diplot.
 - `(-1,5)`: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

5. Fungsi Trigonometri Dua Variabel

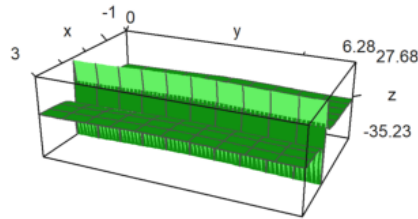
$$g(x,y) = \tan(x) - \cot(y)$$

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



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

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

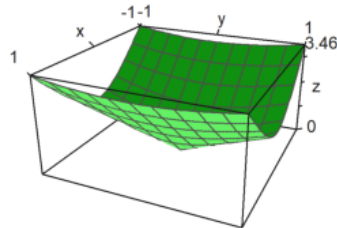


- Perintah `plot3d("g(x,y)",-1,3,0,2*pi,frame=5,zoom=3)` adalah perintah untuk membuat plot tiga dimensi dari fungsi 'g(x,y)' dengan batas 'x' dari '-1' hingga '3' dan batas 'y' dari '0' hingga '2pi'.
 - `plot3d`: perintah untuk membuat plot tiga dimensi.
 - `"g(x,y)"`: fungsi yang akan diplot.
 - `(-1,3)`: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

6. Fungsi Akar Kuadrat

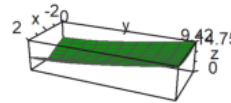
$$P(x, y) = \sqrt{10x^2 + 2y^2}$$

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



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

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



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

Menggambar Data x, y, z * pada ruang Tiga Dimensi (3D)

Definisi

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

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

Tujuan:

Tujuan dari menggambar data 3D adalah untuk membantu pemahaman dan

interpretasi data yang lebih baik, terutama ketika data tersebut memiliki komponen yang tidak dapat direpresentasikan dengan baik dalam dua dimensi.

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

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

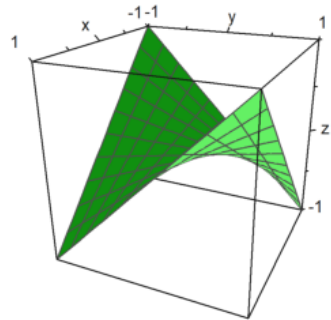
Karena x,y,z adalah matriks, kita asumsikan bahwa (t,s) melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

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

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

Contoh 1

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



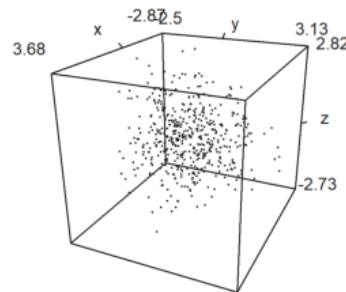
Baris pertama kode "t=-1:0.1:1" membuat vektor baris t yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Baris kedua "s=(-1:0.1:1)'" membuat vektor kolom s yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Operator transpose ' digunakan untuk mengubah vektor baris t menjadi vektor kolom.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

Contoh 2

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

Gayanya sama seperti di plot2d dengan points=true;

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



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

Contoh 3

Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan bayangan dari bola yang

terdistorsi. Koordinat biasa untuk bola adalah

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

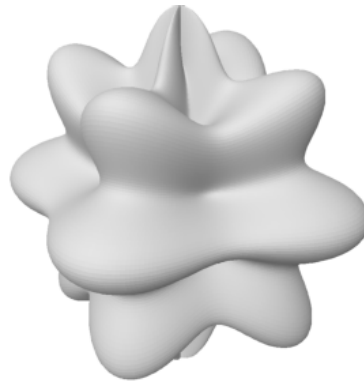
dengan

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

Kami mendistorsi ini dengan sebuah faktor

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

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



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

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

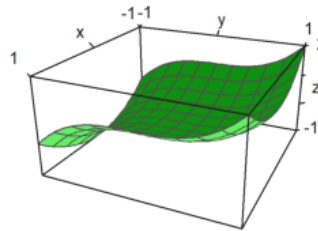
Grafik Tiga Dimensi yang * Bersifat Interaktif dan animasi grafik

3D

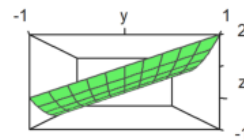
Membuat gambar grafik tiga dimensi (3D) yang bersifat interaktif dan animasi grafik 3D adalah proses menciptakan visualisasi tiga dimensi yang memungkinkan pengguna berinteraksi dengan objek-objek 3D. Interaktivitas dalam gambar 3D memungkinkan pengguna untuk melakukan tindakan seperti mengubah sudut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

CONTOH GAMBAR

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

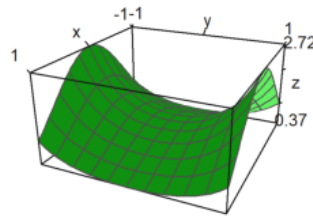
```
>function testplot () := plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0); ...
>rotate("testplot"); testplot():
```



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

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

Turn with the vector keys (press return to finish)

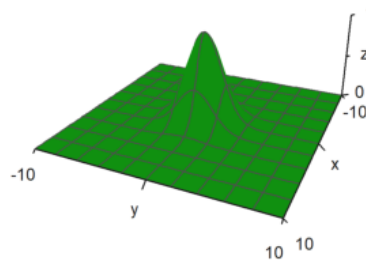


```
>plot3d("exp(x^2+y^2)",>user, ...  
>title="Coba gerakan")
```

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

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

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



Parameter "r=10" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 10.

Parameter "n=80" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.

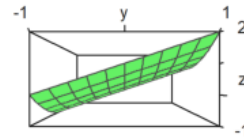
Parameter "fscale=4" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 4.

Parameter "scale=1.2" menunjukkan faktor skala pada plot. Semakin besar nilai scale, semakin besar

ukuran plot yang dihasilkan.

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

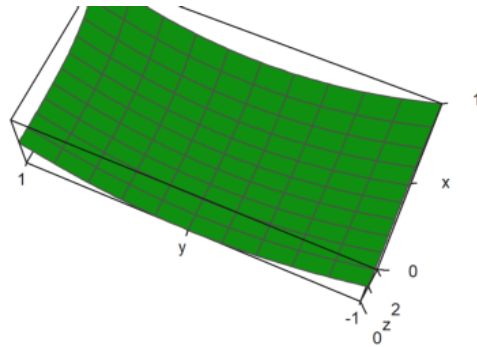
```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Tampilan dapat diubah dengan berbagai cara.

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

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



Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter `center`.

Parameter `center` digunakan untuk memindahkan pusat plot ke lokasi tertentu dalam ruang. Dalam hal ini, pusat plot diatur ke titik (0.4, 0, 0) dalam ruang tiga dimensi. Parameter `center` berguna ketika kita ingin mengubah sudut pandang plot atau ketika kita ingin menyelaraskan plot dengan objek lain dalam scene. Dengan menentukan pusat plot, kita dapat mengontrol posisi kamera dan arah tampilan plot.

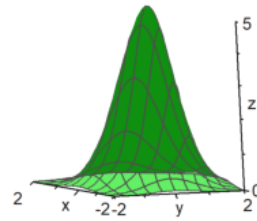
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

`fscale`: menskalakan ke nilai fungsi (defaultnya adalah `<fscale>`).

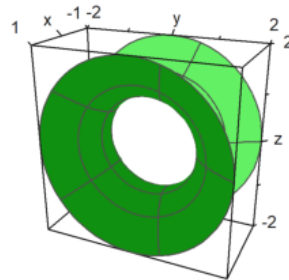
`scale`: angka atau vektor 1x2 untuk diskalakan ke arah x dan y.

`frame`: jenis bingkai (default 1).

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



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



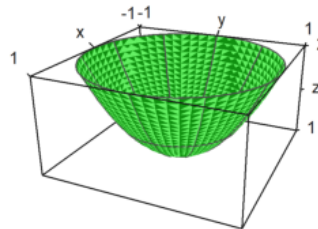
Penjelasan:

Secara umum, parameter "a" dan "b" digunakan untuk menentukan rentang nilai variabel independen dalam suatu fungsi. Dalam kasus ini, "a=-1" dan "b=1" menunjukkan bahwa fungsi tersebut akan diplot pada interval [-1, 1]. Parameter "rotate=true" menunjukkan bahwa grafik akan diputar untuk memberikan tampilan bentuk tiga dimensi yang lebih baik. Parameter "grid=5" menunjukkan bahwa grid dengan jarak 5 unit akan ditampilkan pada grafik.

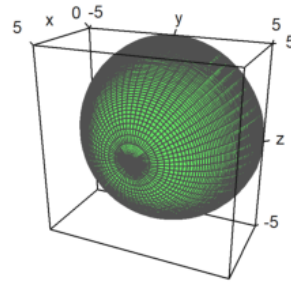
Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

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

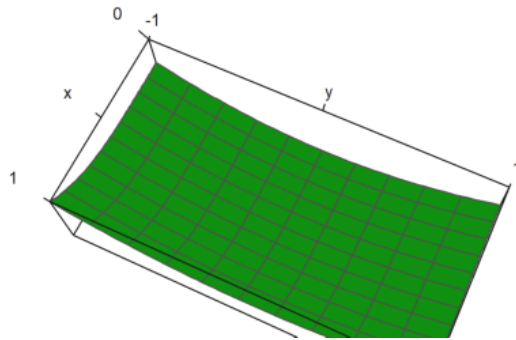
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



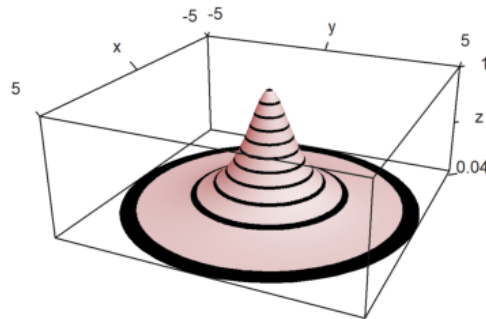
```
>function testplot () := plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1); ...  
>rotate("testplot"); testplot():
```



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




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



Parameter "r=5" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 5.

Parameter ">polar" menunjukkan bahwa plot yang dibuat adalah plot polar tiga dimensi. Plot polar adalah plot yang dibuat dengan menggunakan koordinat polar, yaitu koordinat yang terdiri dari jarak dan sudut.

Parameter "fscale=2" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 2.

Parameter ">hue" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.

Parameter "n=100" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.

Parameter "zoom=4" menunjukkan level zoom pada plot.

Parameter ">contour" menunjukkan bahwa garis kontur akan ditampilkan pada plot.

Parameter "color=blue" menunjukkan warna garis kontur pada plot. Dalam hal ini, warna yang digunakan adalah biru.

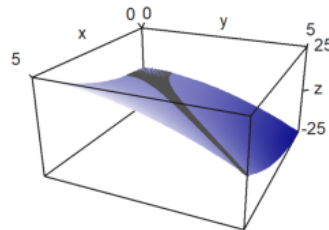
Untuk plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis level dan satu warna atau warna spektral. Euler dapat menggambar ketinggian fungsi pada sebuah plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

-hue: Mengaktifkan bayangan cahaya, bukan kabel.

-contour: Membuat plot garis kontur otomatis pada plot.

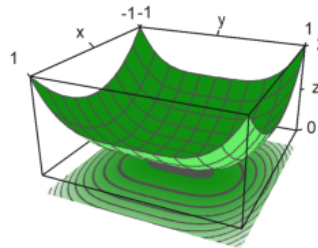
-level=... (atau level): Vektor nilai garis kontur.

```
>function testplot () := plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=blue); ...  
>rotate("testplot"); testplot():
```



Parameter "level=-1:0.1:1" menunjukkan rentang nilai fungsinya yang akan ditampilkan pada plot. Dalam hal ini, rentang nilai fungsinya adalah dari -1 hingga 1 dengan interval 0.1.

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



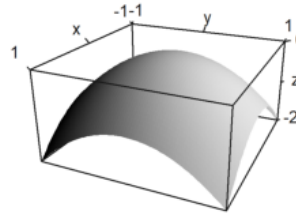
Parameter ">cp" menunjukkan bahwa titik kontrol akan ditambahkan pada plot. Titik kontrol digunakan untuk menentukan bentuk dan posisi plot tiga dimensi.

Parameter "cpcolor=green" menunjukkan warna titik kontrol yang akan digunakan. Dalam hal ini, warna yang digunakan adalah hijau.

Parameter "cpdelta=0.2" menunjukkan jarak antara titik kontrol. Semakin kecil nilai cpdelta, semakin banyak titik kontrol yang akan ditambahkan pada plot.

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

Press I and cursor keys (return to exit)

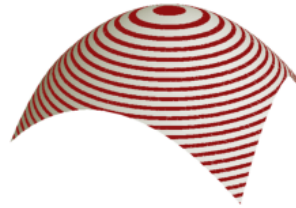


parameter "hue=true" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.

Parameter "light=light=[0,1,1]" menunjukkan intensitas cahaya pada plot. Nilai light=[0,1,1] menunjukkan bahwa cahaya datang dari arah positif y dan z.

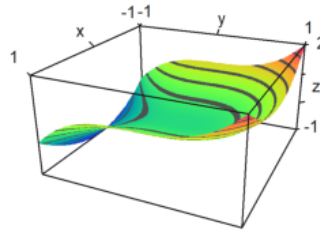
Parameter "amb=0" menunjukkan intensitas cahaya ambient pada plot. Nilai 0 menunjukkan bahwa tidak ada cahaya ambient yang digunakan.

```
>function testplot () := plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=3,contourcolor=red,level=-2:0.1:1,d1=0.01); ...  
>rotate("testplot"); testplot():
```

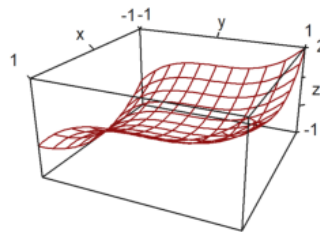


Parameter "frame=false" digunakan untuk menghilangkan frame pada plot tiga dimensi. Parameter "color=rgb(0.2,0.2,0)" menunjukkan warna dasar plot. Dalam hal ini, warna yang digunakan adalah hitam dengan nilai RGB (0.2, 0.2, 0). Parameter "dl=0.01" menunjukkan jarak antara titik-titik pada plot. Semakin kecil nilai dl, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat. Namun, semakin kecil nilai dl, semakin lama waktu yang dibutuhkan untuk membuat plot.

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



```
>function testplot () := plot3d("x^2+y^3", >transparent, grid=10, wirecolor=red); ...  
>rotate("testplot"); testplot():
```

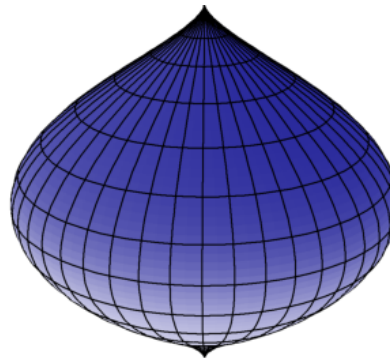


Fungsi Parametrik 3D

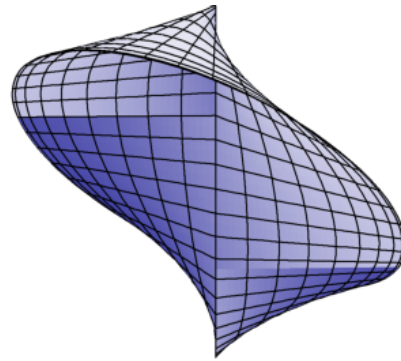
Fungsi parametrik merupakan jenis fungsi matematika yang menggambarkan hubungan antara dua atau lebih variabel, dimana masing-masing koordinat (x , y , z ...) dinyatakan sebagai fungsi lain dari beberapa parameter. Fungsi parametrik dapat digunakan untuk menggambar kurva, lintasan, atau hubungan antara berbagai variabel yang bergantung pada parameter-parameter tertentu.

Sebagai contoh :

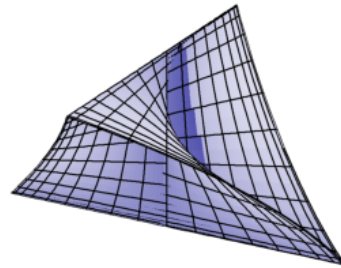
```
>plot3d("cos(x)*cos(y)^3","sin(x)*cos(y)^3","sin(y)", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,3],<frame,...  
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



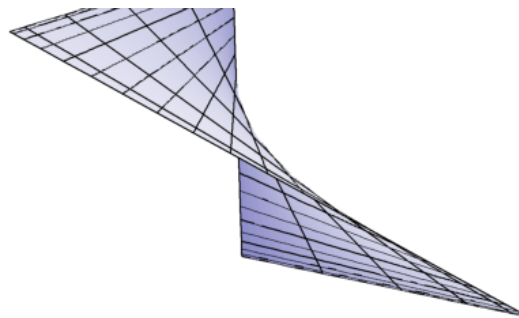
```
>plot3d("cos(x)*cos(y)","sin(x)*cos(y)","cos(x)", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,3],<frame,...  
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



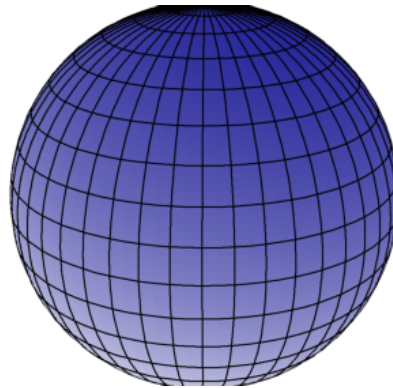
```
>plot3d("cos(x)^3*sin(y)","sin(x)^2*sin(y)","cos(x)^2", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,5],<frame,...  
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```

```
>plot3d("cos(x)^2*cos(y)","sin(x)^2*cos(y)","cos(x)^2", a=0,b=2*pi,c=pi/2,d=-pi/2,...
>>hue,color=blue,light=[0,1,5],<frame,...
>n=90,grid=[10,50],wirecolor=black,zoom=5):
```



```
>plot3d("cos(x)*cos(y)","sin(x)*cos(y)","sin(y)", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,3],<frame,...  
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



Menggambar Fungsi Implisit Implisit

Fungsi implisit (implicit function) adalah fungsi yang memuat lebih dari satu variabel, berjenis variabel bebas dan variabel terikat yang berada dalam satu ruas sehingga tidak bisa dipisahkan pada ruas yang berbeda.

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

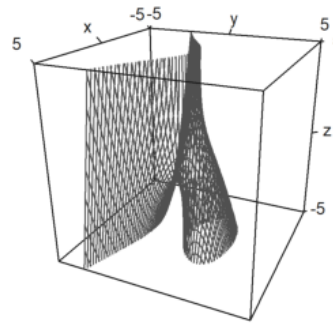
(1 persamaan dan 3 variabel), terdiri dari 2 variabel bebas dan 1 terikat

Misalnya,

$$F(x, y, z) = x^2 + y^2 + z^2 = 1$$

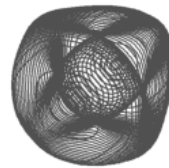
adalah persamaan implisit yang menggambarkan bola dengan jari-jari 1 dan pusat di (0,0,0).

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

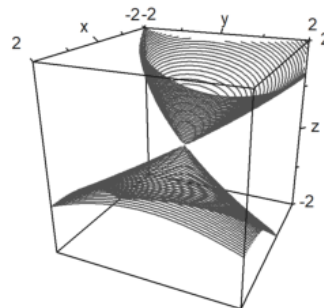


```
>c=1; d=1;
```

```
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d", r=2
```



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



Selain plot kontur yang sudah di jelaskan sebelumnya, pada EMT juga ada plot umplisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel.

Solusi dari

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

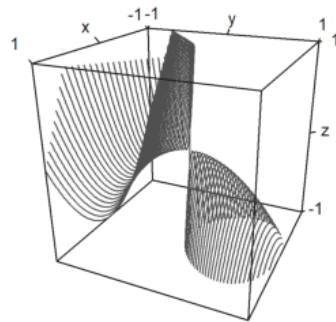
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

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

Ambil contoh dari persamaan latex pada fungsi implisit tadi dan tambahkan nilai-nilai ini, sehingga kita dapat memplot persamaan ini

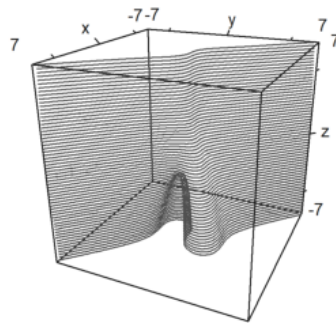
$$M = (x, y, z) : x^2 + y^3 + zy = 1$$

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

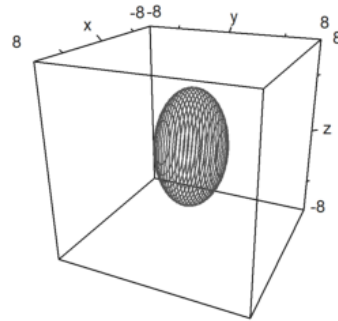


Contoh fungsi implisit yang lainnya

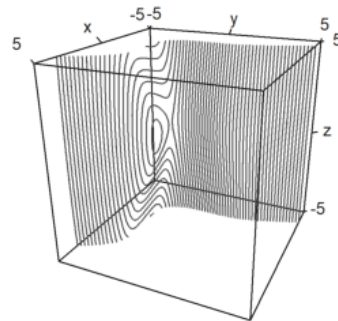
```
>plot3d("x^3+y^3+z*y-1",r=7,implicit=4):
```



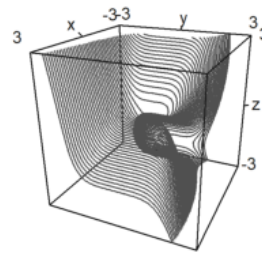
```
>plot3d("2*x^2 + 3*y^2 + z^2 - 25",r=8,implicit=2):
```



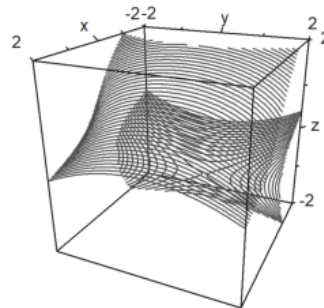
```
>plot3d("x^5 + 5*y^3 + 3*z^2 - 5*x - 7*y - 5*z + 10",r=5,implicit=2):
```



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



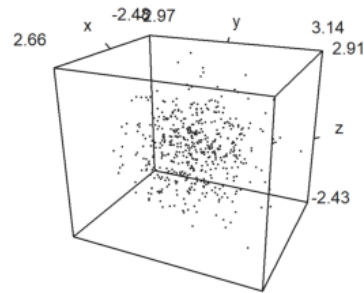
```
>plot3d("x^2+y^2+4*x*z+z^3-5",>implicit,r=2,zoom=2.5):
```



Menggambar Titik pada Ruang Tiga Dimensi

Untuk menggambar titik pada ruang tiga dimensi kita memerlukan tiga vektor untuk koordinat titik serta menambahkan parameter `points=true`.

```
>n=510; ...  
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```



```
>n=510; ...
```

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

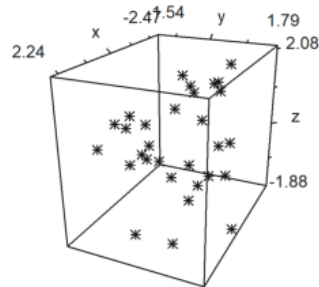
- plot3d() untuk menjalankan perintah membuat plot 3D.

- normal(1,n) sebagai titik koordinat yang akan diplot pada sumbu x,y,z dengan nilai angka distribusi normal yang dicetak secara random sebanyak n sehingga membentuk matriks 1xn atau 1x500.

- points=true sebagai parameter yang memerintahkan plot3d akan menampilkan points(titik-titik).

```
>n=30; ...
```

```
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="*"):
```



1. $n = 30$; Ini adalah pernyataan untuk menginisialisasi variabel n dengan nilai 30. Variabel ini kemungkinan akan digunakan untuk menentukan jumlah titik yang akan digunakan dalam plot.

2. $\text{normal}(1, n)$: Fungsi normal digunakan untuk menghasilkan nilai-nilai acak yang terdistribusi secara normal (disebut juga Gaussian) dengan rata-rata 1 dan deviasi standar 1. Ini berarti Anda akan mendapatkan n nilai acak yang terdistribusi secara normal dengan rata-rata 1 dan deviasi standar 1. Anda melakukan ini untuk mendapatkan koordinat x , y , dan z untuk plot 3D.

3. `plot3d(...)`: Ini adalah fungsi yang digunakan untuk membuat plot 3D dengan parameter-parameter berikut:

`-normal(1, n)`: Ini adalah data koordinat x, y, dan z yang telah

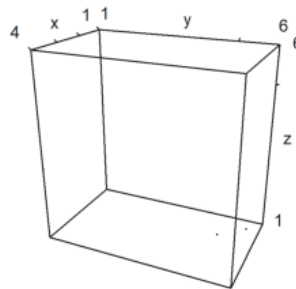
dihasilkan sebelumnya.

`-points = true`: Ini mengatur agar titik-titik data ditampilkan dalam

plot.

`-style = "*"` : Ini mengatur gaya plot menjadi tanda bintang (*).

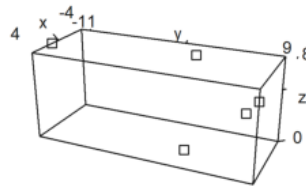
```
>x=[1,2,3,4]; y=[4,5,6,1]; z=[6,1,2,3];  
>plot3d(x,y,z,points=true,style="."):
```



```
>x=[1,2,3,4]; y=[4,5,6,1]; z=[6,1,2,3];  
>plot3d(x,y,z,points=true,style="."):
```

- plot3d() untuk menjalankan perintah membuat plot 3D.
- Titik koordinat yang akan diplot pada sumbu x,y,z telah didefinisikan oleh vektor baris x,y,z sebelumnya.
- points=true sebagai parameter yang memerintahkan plot3d akan menampilkan points(titik-titik).

```
>x=[1,1,0,-4,4]; y=[2,-11,7,1,9]; z=[0,8,4,7,7];  
>plot3d(x,y,z,points=true,zoom=3,style="/"):
```



1.x, y, dan z: Ini adalah tiga vektor yang berisi koordinat titik-titik dalam tiga dimensi. Dalam contoh ini, x berisi [1, 1, 0, -4, 4], y berisi [2, -11, 7, 1, 9], dan z berisi [0, 8, 4, 7, 7]. Setiap elemen dalam vektor-vektor ini mewakili koordinat satu titik dalam ruang 3D.

2.plot3d(...): Ini adalah fungsi yang digunakan untuk membuat plot 3D dengan parameter-parameter berikut:

-x, y, dan z: Ini adalah data koordinat yang akan digunakan untuk membuat plot.

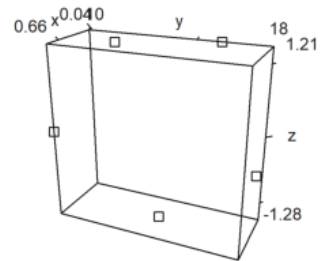
-points = true: Ini mengatur agar titik-titik data ditampilkan dalam plot. Dengan points = true, Anda

akan melihat titik-titik yang mewakili koordinat data.

-zoom = 3: Ini mengatur tingkat zoom plot. Dengan zoom = 3, plot akan diperbesar sebanyak tiga kali dari ukuran defaultnya.

-style = "/" : Ini mengatur gaya plot menjadi tanda garis miring ("/").

```
>a=random(1,5); b=linspace(10,18,4); c=normal(1,5); ...  
>plot3d(a,b,c,scale=[5,1,3],points=true,style="''):
```



```
>a=random(1,5); b=linspace(10,18,4); c=normal(1,5); ...
```

```
plot3d(a,b,c,scale=[5,1,3],points=true,style="''):
```

- `plot3d()` untuk menjalankan perintah membuat plot 3D.
- Titik koordinat yang akan diplot pada sumbu x,y,z didefinisikan oleh `x=a=random(1,5)` yaitu bilangan acak dari 0-1 sebanyak 5 bilangan, `y=b=linspace(10,18,4)` yaitu bilangan dari 10 hingga 18 dengan selisih yang sama sebanyak 5 bilangan, `z=c=normal(1,5)` yaitu bilangan acak distribusi normal sebanyak 5 bilangan.
- `points=true` sebagai parameter yang memerintahkan `plot3d` akan menampilkan points(titik-titik).

Mengatur Tampilan, Warna dan * Angle Gambar Permukaan 3D

Dalam `plot3d` terdapat banyak function terkait tampilan gambar 3D, di antaranya:

`sliced`:

`Memplot versi irisan (0=tidak, 1=arah-x, 2=arah-y).`

`hue` :

`Menghitung bayangan menggunakan sumber cahaya.`

`light, amb, max` :

`Mengontrol pengaturan bayangan titik cahaya, ambient dan maksimum.`

contour :

Menampilkan garis level tebal (dengan level otomatis).

spectral:

Gunakan warna spektral alih-alih rona monokrom. Terdapat skema spektral dari spektral = 1 (> spektral) hingga spektral = 9. >Default >spectral adalah rona warna dan ini setara dengan color=-2 hingga color=-10.

xhue, yhue, zhue:

Gunakan koordinat ini sebagai pengganti sumber cahaya.

hues :

Matriks nilai rona dari 0 sampai 1 untuk bayangan untuk plot x-y-z. Matriks harus memiliki ukuran yang kompatibel dengan x, y, z.

contourcolor :

Warna garis kontur.

contourwidth :

Lebar garis kontur.

fillcolor :

Warna isian untuk permukaan 3d tanpa rona.

user :

Pengguna dapat memutar plot dengan keyboard kiri, kanan, atas, bawah. +,- memperbesar plot. Spasi mengatur ulang plot. Return mengakhiri interaksi pengguna. Tombol a menghasilkan plot anaglyph. Tombol l mengalihkan pergerakan sumber cahaya untuk plot rona. Tombol c menggerakkan plot ke atas, bawah, kiri, atau kanan.

rotate :

Memutar plot sebuah fungsi dalam satu ekspresi dalam x.

anaglyph :

Menghasilkan plot 3d anaglyph (>anaglyph). Plot ini membutuhkan kacamata merah untuk dapat dilihat dengan baik.

viewangle :

Sudut pandang default, diputar di sekitar z-

zoom :

Pembesaran tampilan. Standarnya adalah sekitar 2,6.

view :

Tampilan lengkap, vektor 1x4 yang berisi jarak, zoom, sudut pandang, tinggi pandang.

center :

Vektor ini memindahkan pusat plot. Hal ini diperlukan jika plot tidak dipusatkan di (0,0,0) secara otomatis.

style :

Gaya plot.

color :

Warna untuk objek dan permukaan yang diarsir

wirecolor :

Warna untuk plot kawat

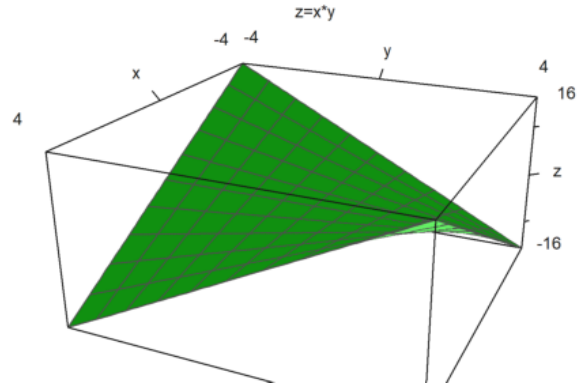
cp :

Menggambar bidang kontur di bawah plot (>cp).

cpcolor :

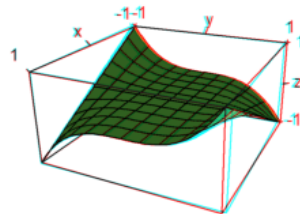
Warna untuk bidang kontur.

```
>plot3d("x*y",r=4,title="z=x*y",zoom=5):
```

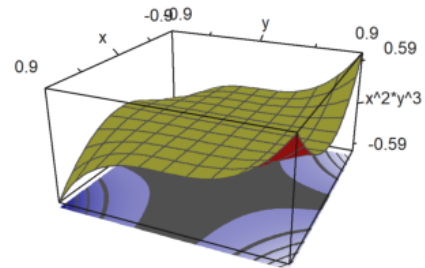


```
>plot3d("x*y^3",>user,r=1,>anaglyph,title="Press cursor keys or return!"):
```

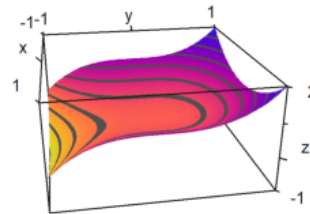
Press cursor keys or return!



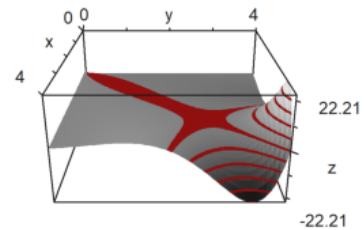
```
>plot3d("x^2*y^3",r=0.9,zlabel="x^2*y^3",>user,zoom=3, ...  
>fillcolor=[2,6],>cp,cpcolor=blue):
```



```
>plot3d("x^2+y^3",angle=80,>contour,spectral=2):
```

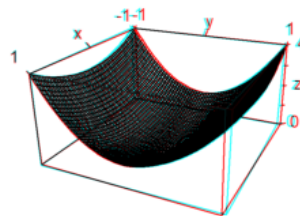


```
>plot3d("x^y-y^x",a=0,b=4,c=0,d=4,angle=90,>contour, ...
> contourwidth=4,contourcolor=red):
```

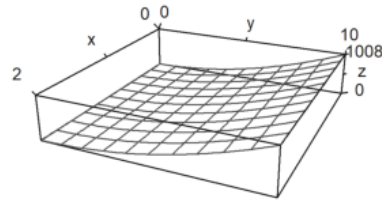


```
>plot3d("x^2+3y^2",>wire,>anaglyph,title="Use Red/Cyan Glasses!"):
```

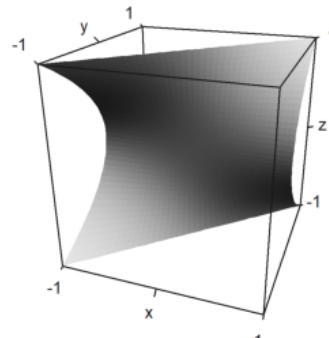
Use Red/Cyan Glasses!



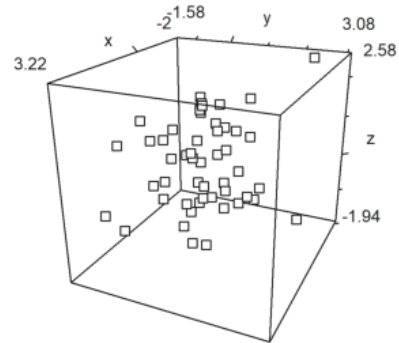
```
>plot3d("x^3+10y^2",0,2,0,10,scale=[5,1,2],zoom=3,grid=10,>transparent):
```



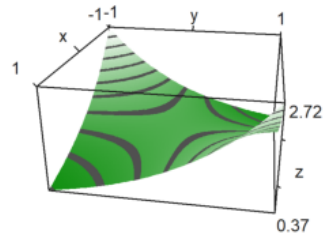
```
>x=-1:0.05:1; y=x'; plot3d(x,x*y^2,y,>user,>hue,>angle=20°):
```



```
>X=normal(3,50); plot3d(X[1],X[2],X[3],>points,style="/",zoom=3,>user):
```

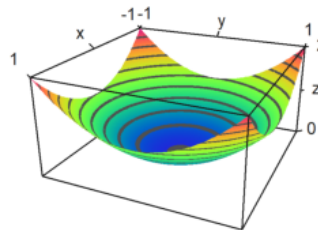


```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```



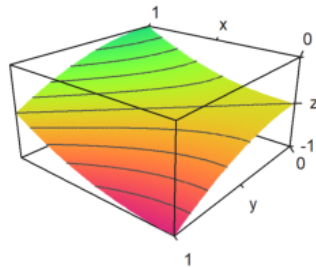
1. `"exp(x*y)"`: Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah eksponensial dari hasil perkalian antara x dan y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. `angle = 100°`: Ini adalah parameter `angle` yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 100 derajat.
3. `contour`: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.
4. `color = green`: Ini adalah parameter `color` yang mengatur warna plot. Dalam hal ini, plot akan menggunakan warna hijau.

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



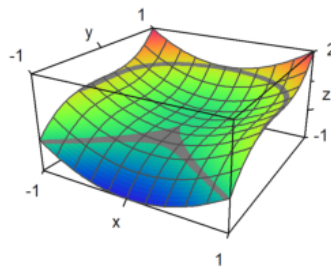
1. $x^2 + y^2$: Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah fungsi kuadrat dari variabel x dan y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. spectral: Ini adalah parameter yang mengatur skema warna plot menjadi skema warna spektral. Dengan menggunakan skema warna spektral, berbagai nilai dalam plot akan diberikan warna yang berbeda, yang memudahkan untuk memahami perubahan nilai dalam fungsi.
3. contour: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.
4. $n = 100$: Ini adalah parameter n yang mengatur jumlah titik sampel dalam plot. Dalam hal ini, ada 100 titik sampel yang akan digunakan untuk menggambarkan plot. Semakin banyak titik sampel, semakin halus plotnya.

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



1. " $x^2 - y^2$ ": Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah perbedaan antara kuadrat variabel x dan kuadrat variabel y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. 0, 1, 0, 1: Ini adalah parameter yang mengatur batasan tampilan plot. Angka-angka ini mewakili batas minimum dan maksimum untuk x dan y . Dalam hal ini, plot akan dibatasi pada rentang x dan y antara 0 dan 1.
3. angle = 220°: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 220 derajat.
4. level = -1:0.2:1: Ini adalah parameter level yang mengatur tingkat nilai fungsi yang akan ditampilkan dalam plot. Rentang ini (-1 hingga 1) akan dibagi menjadi beberapa tingkat, dengan selang 0.2 antara masing-masing tingkat. Ini akan menghasilkan garis kontur pada tingkat nilai fungsi tertentu.
5. color = redgreen: Ini adalah parameter color yang mengatur skema warna plot. Warna yang digunakan adalah kombinasi warna merah dan hijau.

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



1."x² + y³": Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah hasil penjumlahan dari kuadrat variabel x dan kubik variabel y. Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.

2.level = [-0.1, 0.9; 0, 1]: Ini adalah parameter level yang mengatur tingkat nilai fungsi yang akan ditampilkan dalam plot. Parameter ini didefinisikan sebagai matriks dua baris dengan dua kolom. Setiap baris berisi batasan tingkat nilai fungsi yang akan ditampilkan dalam plot. Misalnya, [-0.1, 0.9] menunjukkan bahwa tingkat nilai akan ditampilkan dari -0.1 hingga 0.9, dan [0, 1] menunjukkan bahwa tingkat nilai kedua akan ditampilkan dari 0 hingga 1.

3.spectral: Ini adalah parameter yang mengatur skema warna plot menjadi skema warna spektral.

4.angle = 30°: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 30 derajat.

5.grid = 10: Ini adalah parameter grid yang mengatur jumlah garis kisi dalam plot. Dalam hal ini, akan ada 10 garis kisi dalam plot.

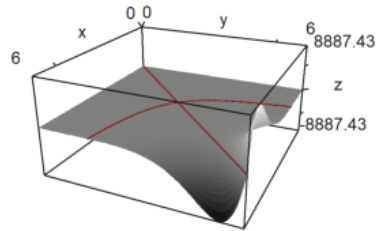
6.contourcolor = gray: Ini adalah parameter contourcolor yang mengatur warna garis kontur dalam plot menjadi abu-abu (gray).

Dalam contoh berikut, kami memplot himpunan, di mana

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

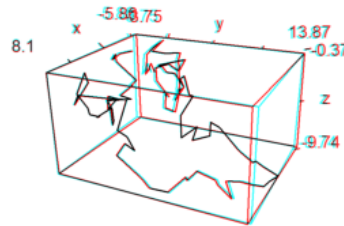
Kami menggunakan satu garis tipis untuk garis level.

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



Menggambar Grafik Tiga Dimensi * alam modulus anaglif

```
>X=cumsum(normal(3,100)); ...  
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



1. `X = cumsum(normal(3, 100));` Ini adalah urutan perintah yang melakukan beberapa operasi berurutan.

a) `normal(3, 100)`: Ini adalah panggilan fungsi normal yang digunakan untuk menghasilkan 100 bilangan acak dengan distribusi normal (Gaussian) dengan rata-rata 3 dan deviasi standar 1. Hasilnya adalah vektor tiga dimensi yang berisi 100 titik acak dalam ruang tiga dimensi.

b) `cumsum(...)`: Ini adalah panggilan fungsi `cumsum` yang digunakan untuk menghitung kumulatif dari vektor 3D yang dihasilkan sebelumnya. Dengan kata lain, ini akan menghasilkan vektor yang merupakan akumulasi (penjumlahan berulang) dari vektor 3D tersebut. Hasilnya adalah vektor tiga dimensi yang menggambarkan perjalanan dalam ruang 3D berdasarkan perubahan titik acak.

2. `plot3d(X[1], X[2], X[3], anaglyph, wire);` Ini adalah perintah untuk membuat plot 3D dari data yang telah dihasilkan sebelumnya.

a). `X[1]`, `X[2]`, dan `X[3]` adalah komponen vektor tiga dimensi `X` yang akan digunakan sebagai koordinat dalam plot 3D. `X[1]` digunakan sebagai koordinat sumbu x, `X[2]` digunakan sebagai koordinat sumbu y, dan `X[3]` digunakan sebagai koordinat sumbu z.

b) `anaglyph`: Ini adalah parameter yang mengatur plot menggunakan efek anaglif. Anaglif adalah teknik yang digunakan untuk menghasilkan efek tiga dimensi (3D) dengan menggunakan dua gambar yang sedikit berbeda untuk mata kiri dan kanan, dan penonton memerlukan kacamata anaglif khusus untuk melihat efek 3D.

c) `wire`: Ini adalah parameter yang mengatur plot sebagai plot tali (wireframe), yang berarti hanya garis-garis yang menghubungkan titik-titik yang akan ditampilkan dalam plot.

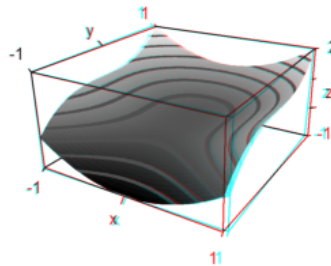
4. `X[1]`, `X[2]`, dan `X[3]`: Ini adalah komponen dari vektor `X`. `X[1]` digunakan sebagai koordinat sumbu

x, X[2] digunakan sebagai koordinat sumbu y, dan X[3] digunakan sebagai koordinat sumbu z. Dengan menggunakan komponen vektor ini sebagai koordinat, Anda membuat plot 3D yang merepresentasikan perubahan dalam tiga dimensi berdasarkan data dalam vektor X.

5.anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif. Efek anaglif adalah teknik yang digunakan untuk menghasilkan efek tiga dimensi (3D) dengan menggunakan dua gambar yang sedikit berbeda untuk mata kiri dan kanan. Penonton memerlukan kacamata anaglif khusus dengan lensa berwarna berbeda untuk mata kiri dan kanan untuk melihat efek 3D ini. Penggunaan anaglyph dalam kode ini menunjukkan bahwa plot akan dibuat dengan efek anaglif.

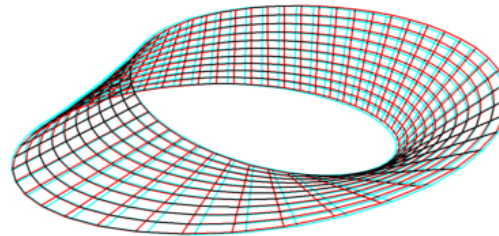
6.wire: Ini adalah parameter yang mengatur plot sebagai plot tali (wireframe). Wireframe adalah gaya plot di mana hanya garis-garis yang menghubungkan titik-titik yang akan ditampilkan dalam plot. Dengan pengaturan wire, Anda akan melihat plot dalam bentuk rangkaian garis-garis yang menggambarkan bentuk objek dalam tampilan 3D.

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



1. $x^2 + y^3$: Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah hasil penjumlahan dari kuadrat variabel x dan kubik variabel y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif. Anaglif adalah teknik yang digunakan untuk menghasilkan efek tiga dimensi (3D) dengan menggunakan dua gambar yang sedikit berbeda untuk mata kiri dan kanan. Penonton memerlukan kacamata anaglif khusus dengan lensa berwarna berbeda untuk mata kiri dan kanan untuk melihat efek 3D ini. Dengan pengaturan ini, plot akan dibuat dengan efek 3D anaglif.
3. contour: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.
4. angle = 30°: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 30 derajat.

```
>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):
```



1. `u = linspace(-1, 1, 10);` Ini adalah perintah untuk membuat vektor `u` yang berisi 10 nilai yang dihasilkan secara merata dalam rentang -1 hingga 1. Vektor ini akan digunakan dalam perhitungan selanjutnya.

2. `v = linspace(0, 2 * pi, 50)';` Ini adalah perintah untuk membuat vektor `v` yang berisi 50 nilai yang dihasilkan secara merata dalam rentang 0 hingga 2π (dua kali nilai π). Vektor ini juga akan digunakan dalam perhitungan selanjutnya.

3. `X`, `Y`, dan `Z`: Ini adalah perintah-perintah yang digunakan untuk menghasilkan data koordinat dalam tiga dimensi. Data ini dihasilkan berdasarkan persamaan yang menggunakan nilai `u` dan `v`. Data ini akan digunakan untuk membuat plot 3D.

4. `plot3d(X, Y, Z, anaglyph, <frame, wire, scale = 2.3);` Ini adalah perintah untuk membuat plot 3D berdasarkan data `X`, `Y`, dan `Z` yang telah dihasilkan sebelumnya. Parameter-parameter yang digunakan dalam perintah ini adalah:

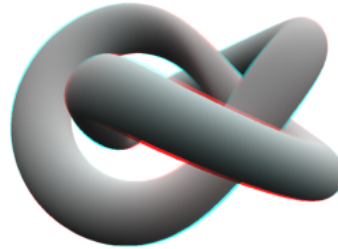
a) `anaglyph`: Ini adalah parameter yang mengatur plot menggunakan efek anaglif, yang memberikan efek tiga dimensi (3D) saat melihat plot dengan kacamata anaglif.

b) `<frame`: Ini adalah parameter yang mengatur agar frame (kerangka) plot ditampilkan. Ini adalah bingkai atau batasan dari plot.

c) `wire`: Ini adalah parameter yang mengatur plot dalam bentuk rangkaian garis-garis (wireframe).

d) `scale = 2.3`: Ini adalah parameter yang mengatur faktor skala plot sebesar 2.3. Ini akan memperbesar plot.

```
>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):
```

1. `u := linspace(-pi, pi, 160);` Ini adalah perintah untuk membuat vektor `u` yang berisi 160 nilai yang dihasilkan secara merata dalam rentang dari $-\pi$ hingga π .
2. `v := linspace(-pi, pi, 400)';` Ini adalah perintah untuk membuat vektor `v` yang berisi 400 nilai yang dihasilkan secara merata dalam rentang dari $-\pi$ hingga π . Vektor ini diubah menjadi matriks kolom dengan penambahan tanda apostrof (`'`) di belakangnya.
3. `x, y, dan z:` Ini adalah perintah-perintah yang digunakan untuk menghasilkan data koordinat dalam tiga dimensi. Data ini dihasilkan berdasarkan persamaan yang menggunakan nilai `u` dan `v`. Data ini akan digunakan untuk membuat plot 3D.
4. `plot3d(x, y, z, frame = 0, scale = 1.5, hue = 1, light = [1, 0, -1], zoom = 2.8, anaglyph);` Ini adalah perintah untuk membuat plot 3D berdasarkan data `x`, `y`, dan `z` yang telah dihasilkan sebelumnya. Parameter-parameter yang digunakan dalam perintah ini adalah:

- a)frame = 0: Ini adalah parameter yang mengatur agar frame (kerangka) plot tidak ditampilkan.
- b)scale = 1.5: Ini adalah parameter yang mengatur faktor skala plot sebesar 1.5. Ini akan memperbesar plot.
- c)hue = 1: Ini adalah parameter yang mengatur warna plot dengan skala warna tunggal.
- d)light = [1, 0, -1]: Ini adalah parameter yang mengatur pencahayaan plot dengan arah cahaya yang ditentukan oleh vektor [1, 0, -1].
- e)zoom = 2.8: Ini adalah parameter yang mengatur tingkat zoom plot sebesar 2.8. Ini akan memperbesar plot.
- f)anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif, yang memberikan efek tiga dimensi (3D) saat dilihat dengan kacamata anaglif.

Plot Statistik batang 3d

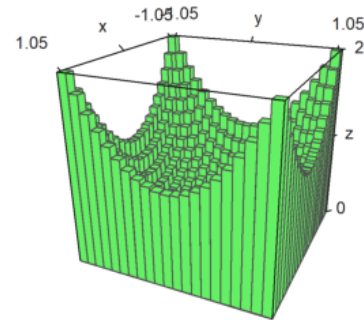
Plot bar juga dimungkinkan. Untuk ini, kita harus menyediakan

- 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.

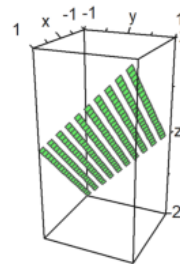
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>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):
```



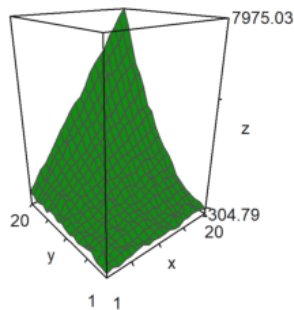
Dimungkinkan untuk membagi plot permukaan menjadi dua atau lebih bagian.

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

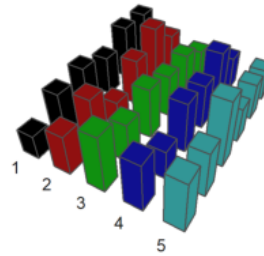


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan `scale(M)`, atau menskalakan matriks dengan `>zscale`. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...  
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```

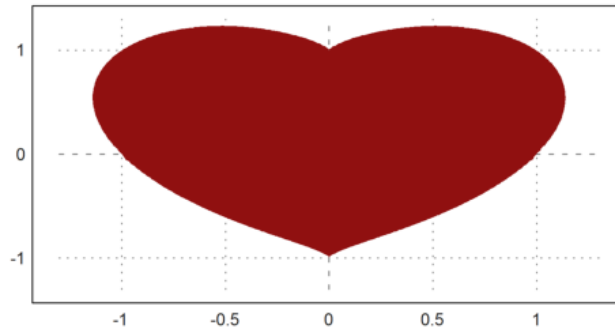


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...  
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...  
>columnplot3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

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



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

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

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut adalah ungkapan, yang mendefinisikan hati:

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

Selanjutnya kita atur

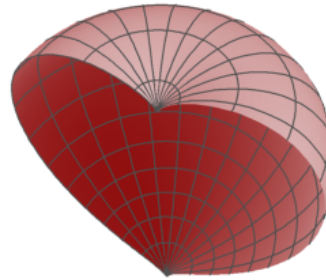
$$x = r.\cos(a), \quad y = r.\sin(a).$$

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

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2\sin a) r^5}{16}$$

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

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

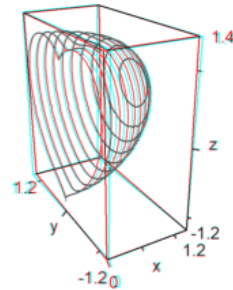


Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>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):
```

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi mengenai materi plot 3D.
- mencari latihan soal di buku dan internet.
- Mempelajari perintah perintah yang ada di EMT berkaitan dengan plot 3D.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam menggunakan perintah terutama saat menggunakan Povray, solusinya dengan mencari informasi di internet

6. Penggunaan software EMT untuk aplikasi Geometri

- a. Hal hal yang dipelajari beserta contohnya
- Memanggil program geometry untuk menggunakan perintah
 - Fungsi fungsi geometri
 - Luas, lingkaran luar, lingkaran dalam segitiga
 - Geometri simbolik
 - Garis dan lingkaran berpotongan.
 - Garis sumbu
 - jarak minimal pada bidang

```
>load geometry
```

Numerical and symbolic geometry.

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

`defaultd:=textheight()*1.5`: nilai asli untuk parameter d
`setPlotrangle(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 vektor v dan w.

`lineThrough(A, B)`: garis melalui A dan B, hasilnya [a,b,c] sdh.

$$ax+by=c.$$

`lineWithDirection(A,v)`: garis melalui A searah vektor v
`getLineDirection(g)`: vektor arah (gradien) garis g
`getNormal(g)`: vektor normal (tegak lurus) garis g
`getPointOnLine(g)`: titik pada garis g
`perpendicular(A, g)`: garis melalui A tegak lurus garis g
`parallel (A, g)`: garis melalui A sejajar garis g
`lineIntersection(g, h)`: titik potong garis g dan h
`projectToLine(A, g)`: proyeksi titik A pada garis g
`distance(A, B)`: jarak titik A dan B
`distanceSquared(A, B)`: kuadrat jarak A dan B
`quadrance(A, B)`: kuadrat jarak A dan B
`areaTriangle(A, B, C)`: luas segitiga ABC
`computeAngle(A, B, C)`: besar sudut $\angle ABC$
`angleBisector(A, B, C)`: garis bagi sudut $\angle ABC$
`circleWithCenter (A, r)`: lingkaran dengan pusat A dan jari-jari r
`getCircleCenter(c)`: pusat lingkaran c
`getCircleRadius(c)`: jari-jari lingkaran c
`circleThrough(A,B,C)`: lingkaran melalui A, B, C
`middlePerpendicular(A, B)`: titik tengah AB
`lineCircleIntersections(g, c)`: titik potong garis g dan lingkaran c
`circleCircleIntersections (c1, c2)`: titik potong lingkaran $c1$ dan

$c2$

`planeThrough(A, B, C)`: bidang melalui titik A, B, C

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan

y dengan titik A pada sisi positif (kanan/atas) garis

quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni

$\sin(\alpha)^2$ dengan α 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 membentuk

suatu segitiga

doublespread(sa): Spread sudut rangkap Spread 2ϕ , dengan

$sa = \sin(\phi)^2$ spread a.

Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tetapkan tiga poin dan plot mereka.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik  
>B=[0,1]; plotPoint(B,"B");  
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB  
>plotSegment(B,C,"a"); // a=BC  
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri meliputi fungsi untuk membuat garis dan lingkaran. Format garis adalah $[a,b,c]$, yang mewakili garis dengan persamaan $ax+by=c$.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1, 2, 2]
```

Hitunglah garis tegak lurus yang melalui A pada BC.

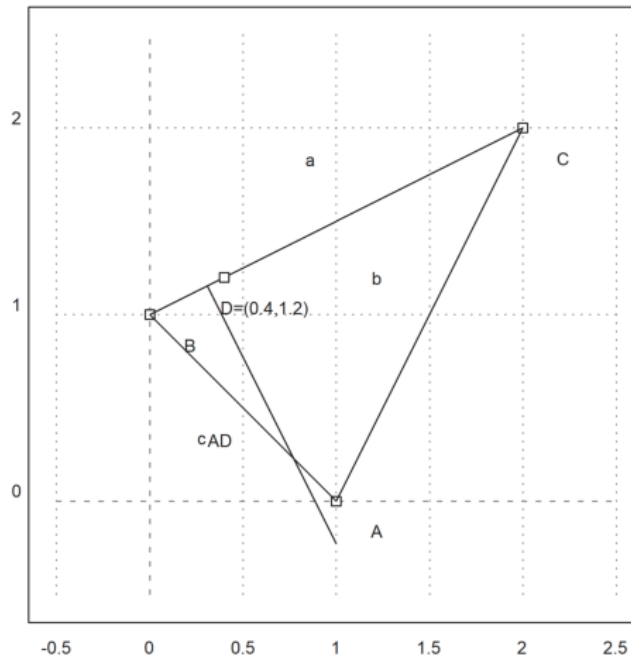
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD \cdot BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D) , BC=norm(B-C)
```

1.5

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D)*distance(B,C)/2
```

1.5

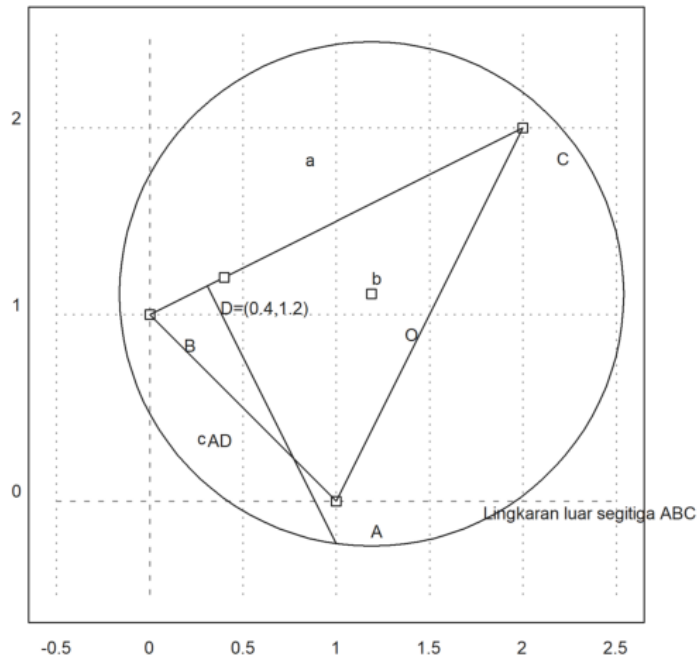
Sudut di C

```
>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.

>0, R

[1.1667, 1.1667]
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
```

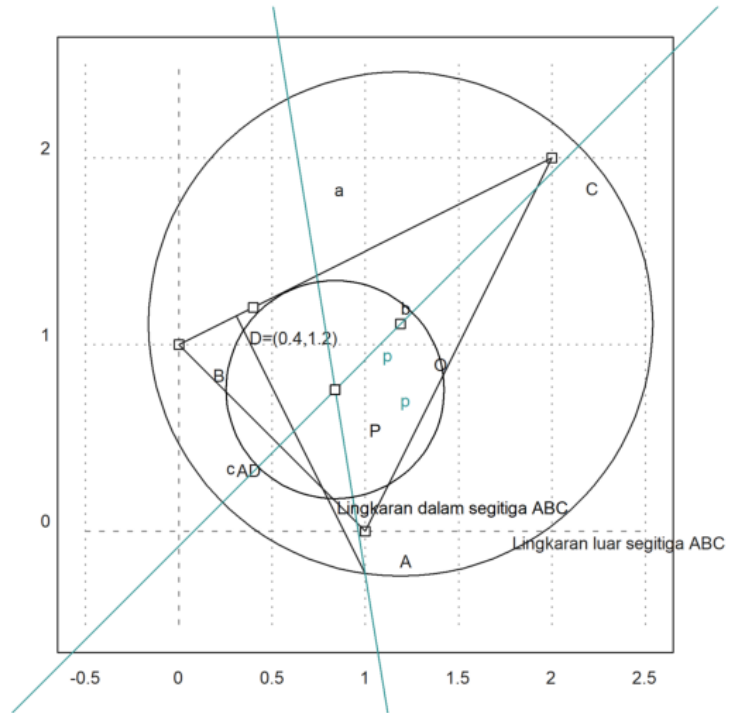
```
[0.86038, 0.86038]
```

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
```



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
```

$$\left[y = \frac{x}{2} + 1 \right]$$

$$\left[y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1) y - x y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

$$[2, 1, 2]$$

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{bmatrix} 2 & 6 \\ - & - \\ 5 & 5 \end{bmatrix}$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\begin{bmatrix} 2 & 6 \\ \frac{2}{5} & \frac{6}{5} \end{bmatrix}$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

1.178511301977579

```
>$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 s
```

$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>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.

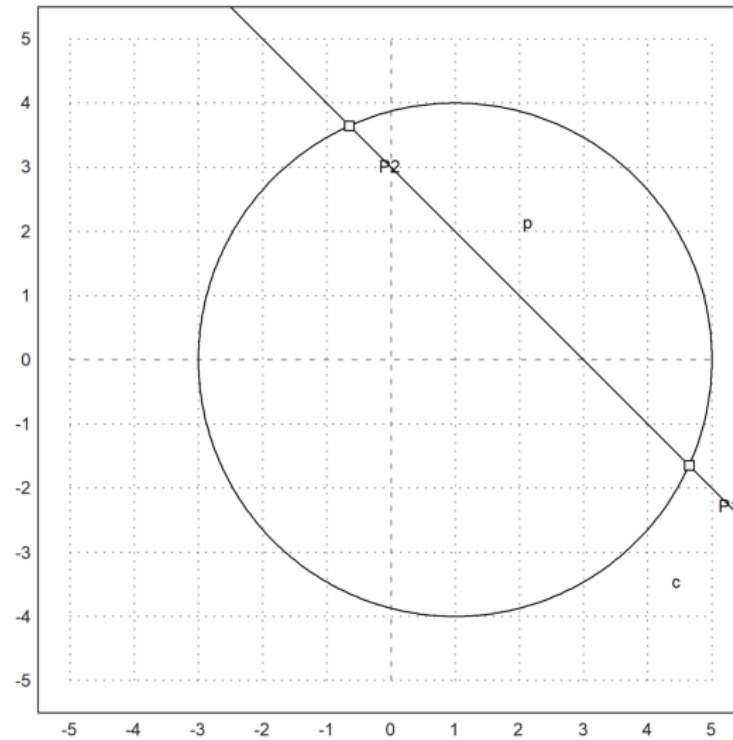
```
>A &:= [1,0]; c=circleWithCenter(A,4);  
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);  
>setPlotRange(5); plotCircle(c); plotLine(l);
```

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 busur yang sama adalah sama besar.

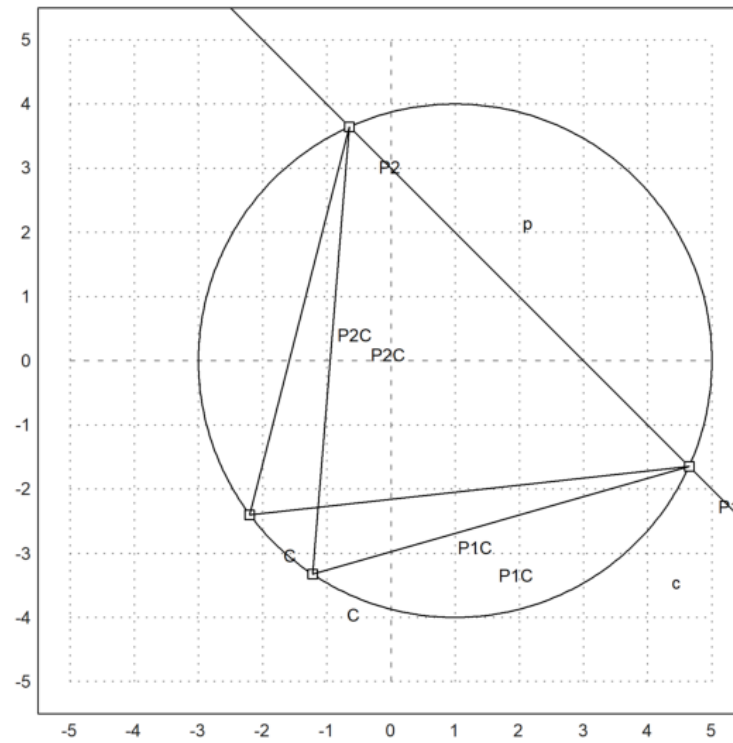
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degprint(computeAngle(P1,C,P2))
```

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''

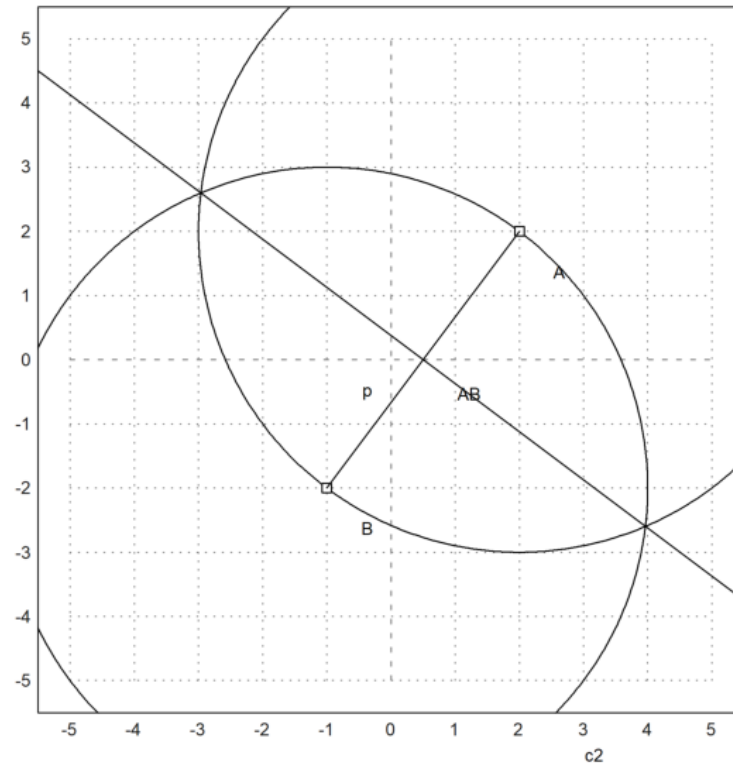
```
>insimg;
```



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 melalui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];  
>c1=circleWithCenter(A,distance(A,B));  
>c2=circleWithCenter(B,distance(A,B));  
>{P1,P2,f}=circleCircleIntersections(c1,c2);  
>l=lineThrough(P1,P2);  
>setPlotRange(5); plotCircle(c1); plotCircle(c2);  
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```



Selanjutnya, kami melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk persimpangan cukup terlibat. Tetapi kita dapat menyederhanakannya, jika kita memecahkan y .

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);  
>$solve(g,y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);  
>$solve(h,y)
```

$$\left[y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

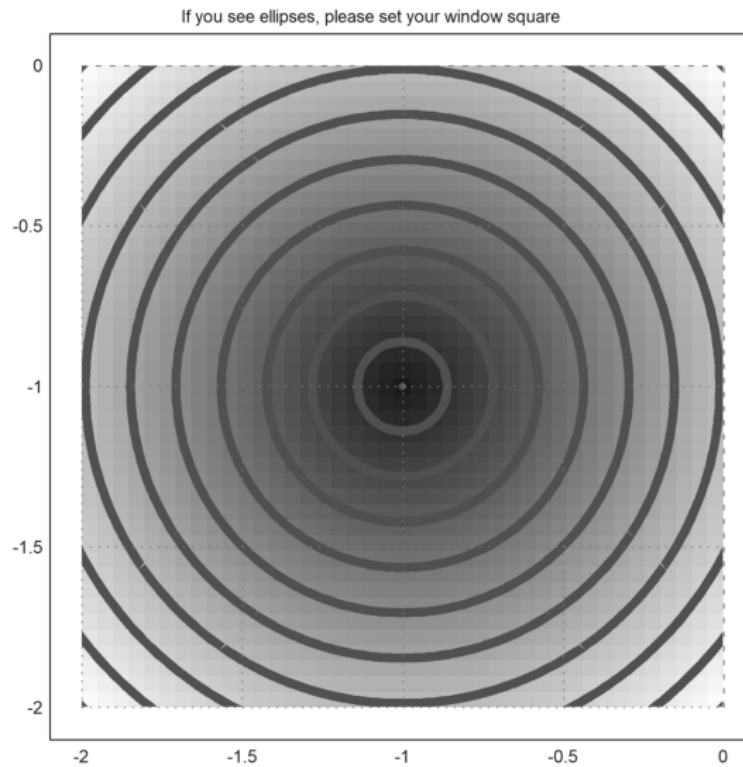
$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

Jarak Minimal pada Bidang

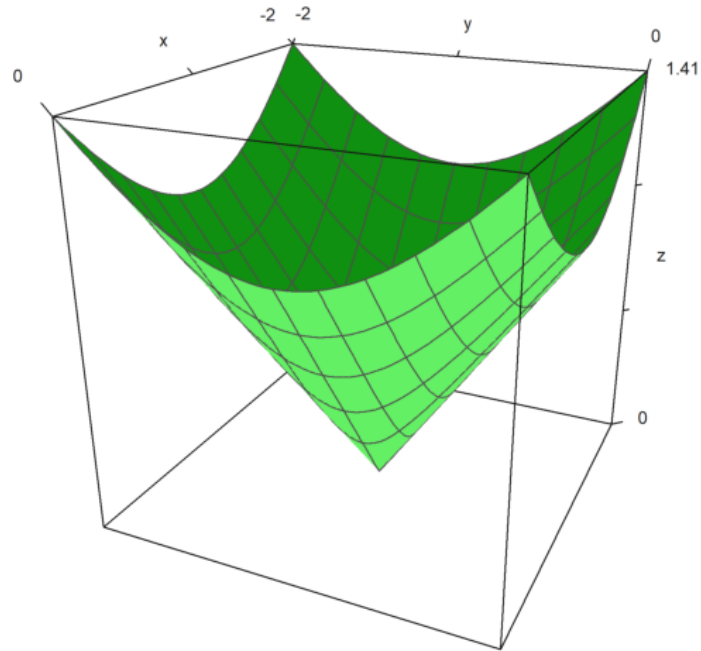
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

```
>&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):
```



b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi mengenai materi geometri.
- mencari latihan soal di buku dan internet.
- Mempelajari perintah perintah yang ada di EMT berkaitan dengan geometri.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam memahami perintah yang berkaitan dengan geometri, solusinya dengan menonton youtube dan mempelajari rumus rumus geometri.

7. Penggunaan software EMT untuk aplikasi Kalkulus

-
- a. Hal hal yang dipelajari beserta contohnya
- 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).

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 beberapa contoh cara mendefinisikan fungsi.

```
>function f(x) &= 2*E^x // fungsi simbolik
```

$$x^2 - 2$$

```
>function g(x) {return 3*x+1}
```

$$3x + 1$$

```
>function h(x) {return f(g(x))} // komposisi fungsi
```

$$x^2 - 2$$

1. Untuk fungsi

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

tentukan nilai

a. $k(-4)$

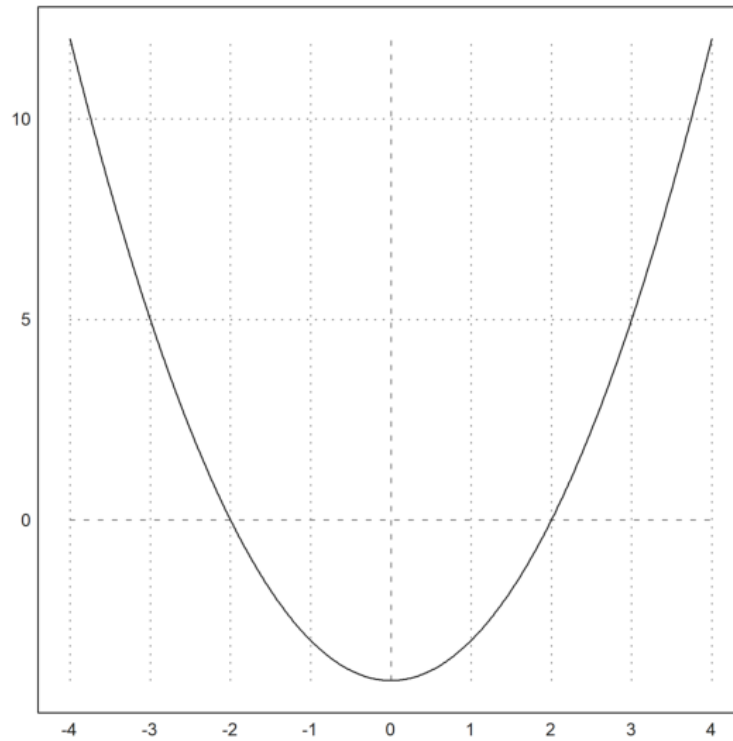
b. $k(4)$

```
>function k(x) := x^2 -4  
>k(-4), k(4)
```

12

12

```
>plot2d("k",-4,4):
```



2. Untuk fungsi

$$z(x) = \frac{x^2 - 16}{x - 4}$$

hitunglah masing-masing nilai.

a. $z(6)$

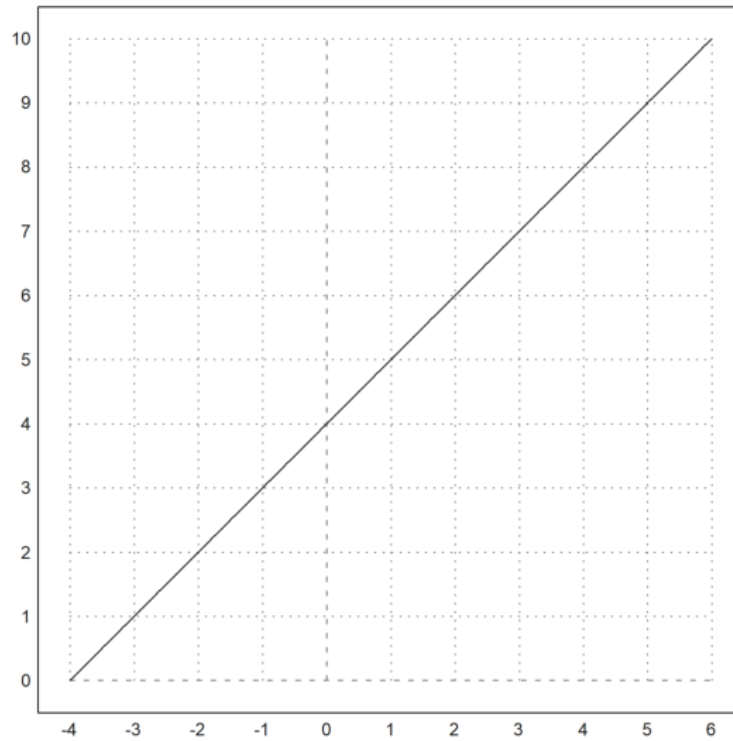
b. $z(2)$

```
>function z(x) := (x^2-16)/(x-4)
>z(6), z(2)
```

10

6

```
>plot2d("z",-4,6):
```



3. Untuk fungsi

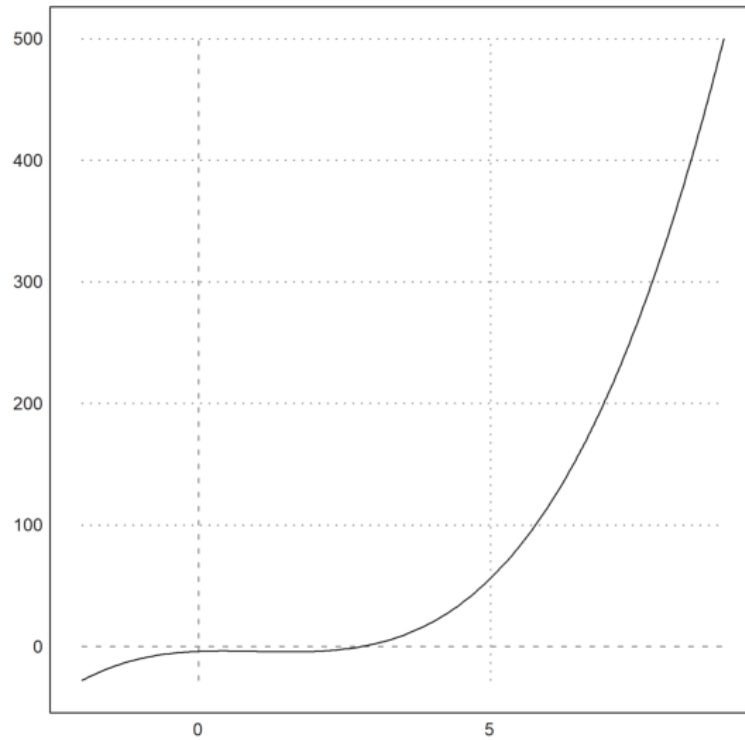
$$r(x) = x^3 - 3x^2 + 2x - 4$$

tentukan nilai $r(4)$, $r(-6)$, $r(8)$

```
>function f(x) := x^3-3*x^2+2*x-4  
>f(4), f(-6), f(8)
```

```
20  
-340  
332
```

```
>plot2d("x^3-3*x^2+2*x-4",-2,9):
```



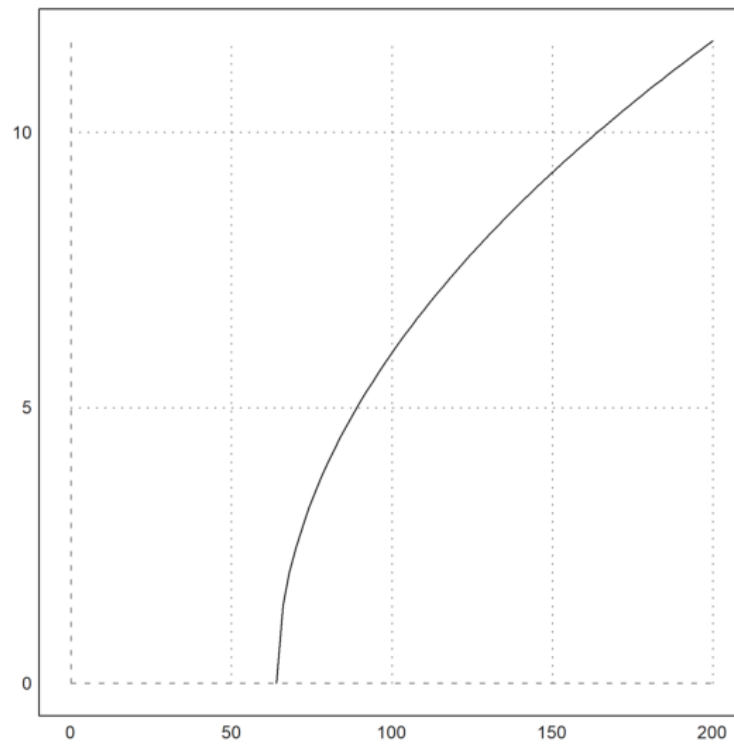
4. Tentukan nilai $f(200)$ dari fungsi berikut

$$f(x) = \sqrt{x - 64}$$

```
>function f(x) := sqrt(x-64)
>f(200)
```

11.6619037897

```
>plot2d("sqrt(x-64)",0,200):
```



5. Untuk fungsi

$$f(x) = x^2 - 3x + 2$$

dan

$$g(x) = x + 3$$

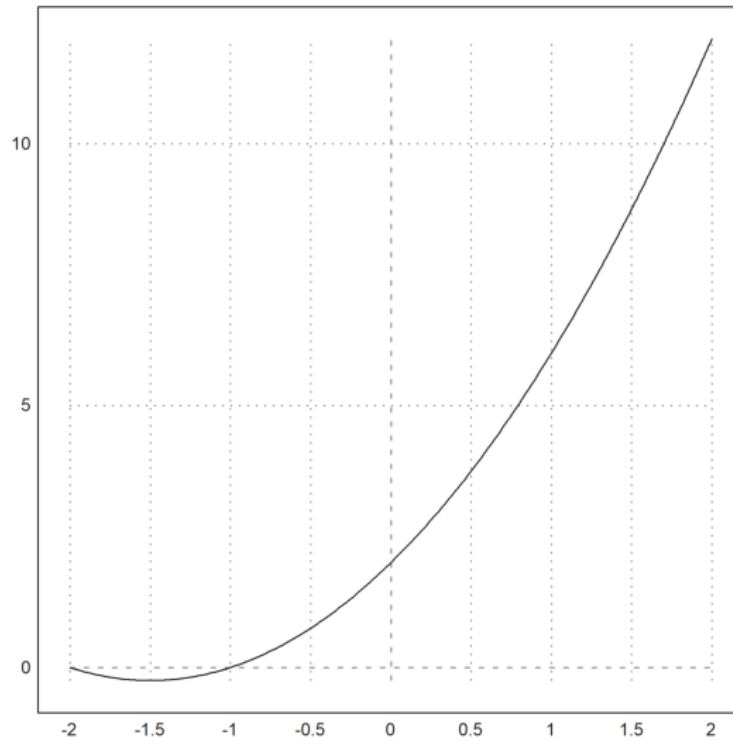
cari nilai $f \circ g(-4)$, $g \circ f(0)$

```
>function f(x) := x^2-3*x+2; $f(x)
>function g(x) := x+3; $g(x)
>f(g(-4)), g(f(0))
```

6

5

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



6. Tentukan nilai dari

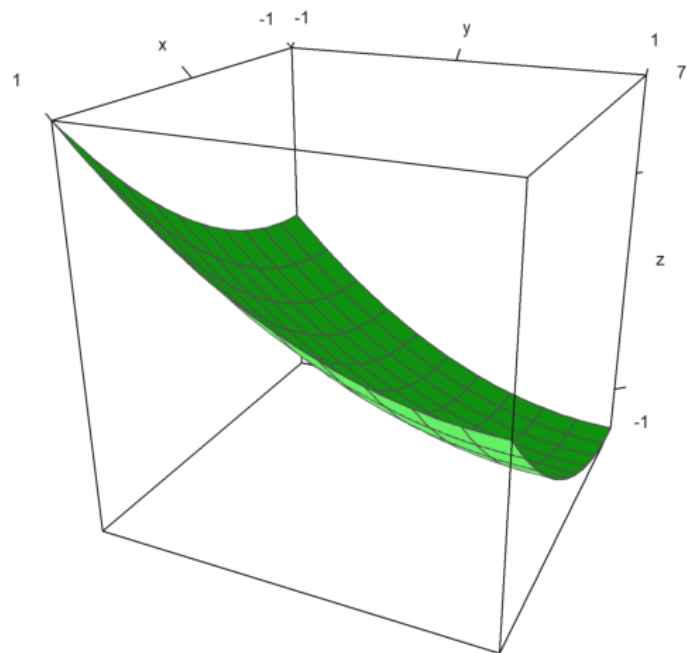
$$f(x, y) := x^2 + y^2 + 2x - 2y + 1$$

dengan $x=1$ dan $y=3$

```
>function f(x,y):= x^2+y^2+2*x-2*y+1  
>f(1,3)
```

7

```
>plot3d("x^2+y^2+2*x-2*y+1"):
```

Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x \rightarrow 0} \frac{1}{2x - 1} = -1$$

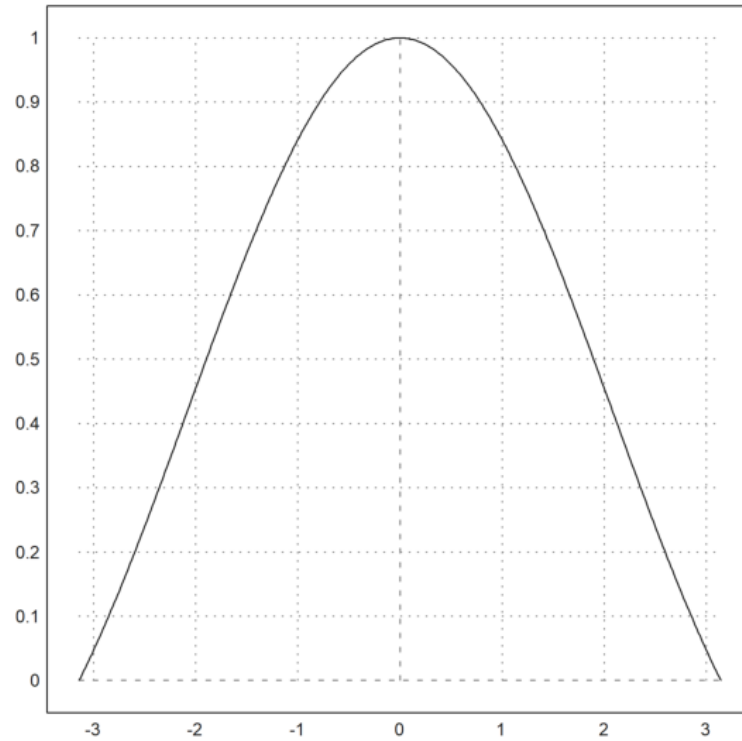
```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi):
```



```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \textit{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \textit{infinity}$$

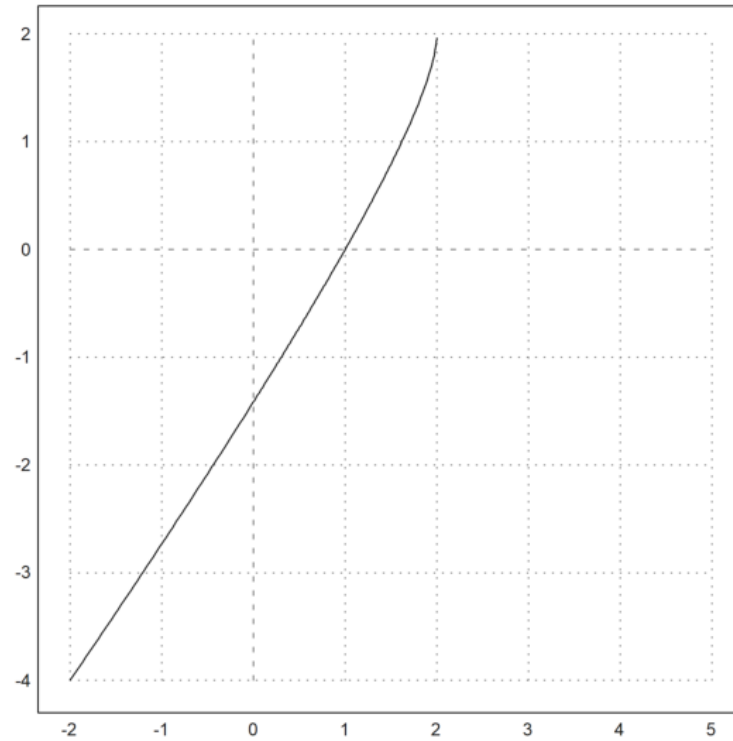
```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3}i$$

```
>plot2d("x-sqrt(2-x)",-2,5):
```



```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left(\frac{1}{x} + 1 \right)^x = e$$

```
>$showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left(\frac{k}{x} + 1 \right)^x = e^k$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x \rightarrow 0} (x+1)^{\frac{1}{x}} = e$$

```
>$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left(\frac{x}{x+k} \right)^x = e^{-k}$$

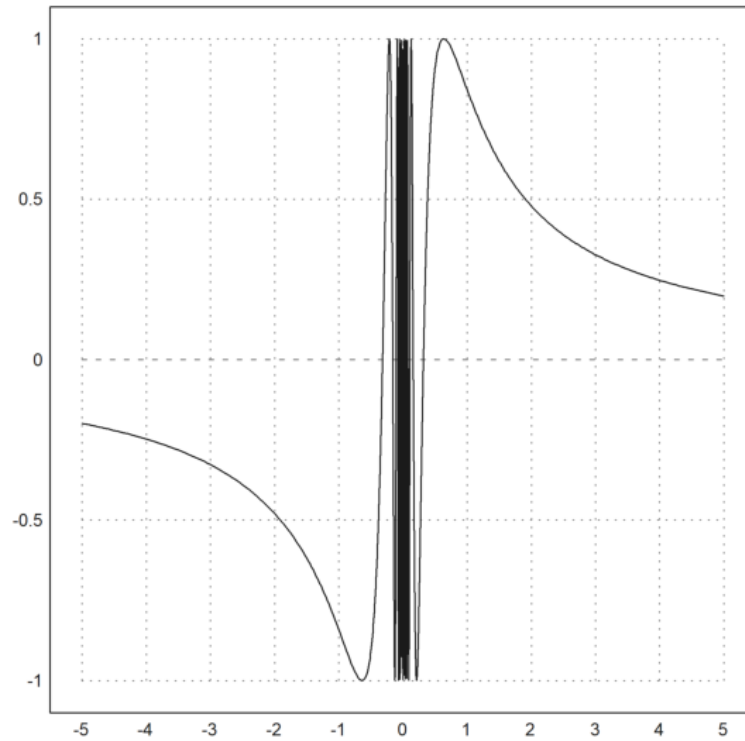
```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin \left(\frac{1}{x} \right) = ind$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin \left(\frac{1}{x} \right) = 0$$

```
>plot2d("sin(1/x)",-5,5):
```



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

x
x


```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

$[x > 0]$

```
>&forget(x<0)
```

$[x < 0]$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

$\sinh(x)$

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).

$$\int_a^b f(x) \, dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

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"

$$\int x^n \, dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1} \, dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2 + 1} dx = \arctan x$$

```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^\pi \sin x dx = 2$$

```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x dx = \cos a - \cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

Answering "Is n positive, negative or zero?" with "positive"

$$\int_a^b x^n dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{2 \cdot 5^{\frac{5}{2}}}{21} - \frac{2}{105}$$

```
>$ratsimp(%)
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{2 \cdot 5^{\frac{7}{2}} - 2}{105}$$

```
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^{\pi} - 1) \sin a + (e^{\pi} + 1) \cos a$$

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

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^{\pi} - 1) (\sin a - \cos a)$$

```
>function map f(x) &= E^(-x^2)
```

$$E^{-x^2}$$

```
>$showev('integrate(f(x),x))
```

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

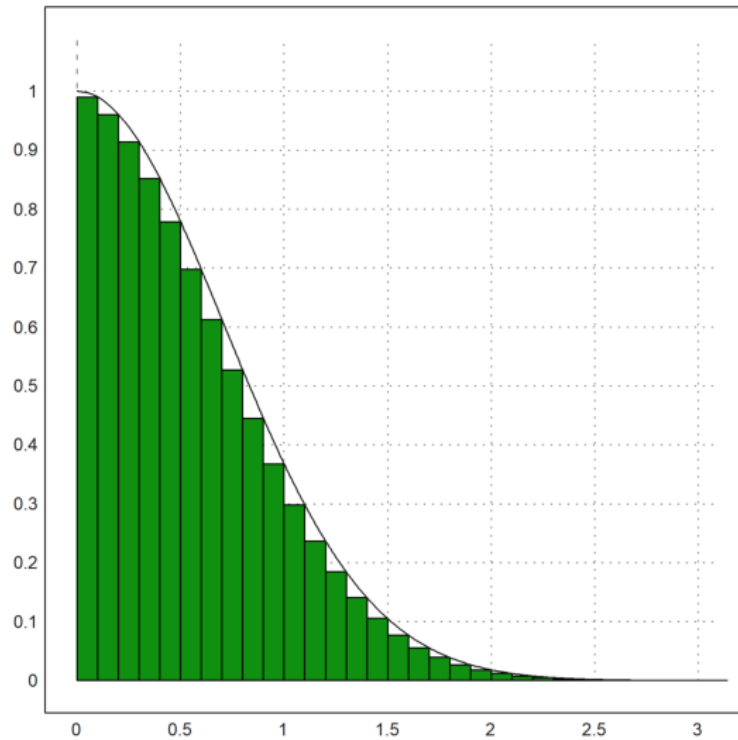
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

$$\int_0^{\pi} x^x dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



Integral tentu

maxima: `'integrate(f(x),x,0,pi)`

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva $y=f(x)$ tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
>// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

```
maxima: 'integrate(f(x),x,0,pi) = 0.1*sum(fx[i],i,1,length(fx))
```

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai $f(x)$ untuk $x = 0.1, 0.2, 0.3, \dots, 3.2$.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

```
0.836219610253
```

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```


$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

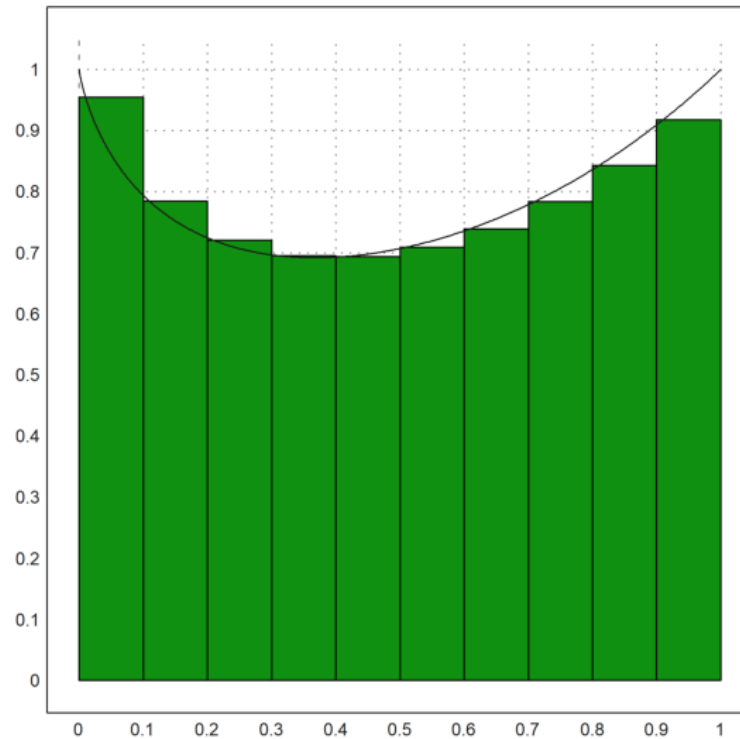
```
>function f(x) &= x^x
```

```
    x
    x
```

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x dx = \int_0^1 x^x dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah `integrate`. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);  
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

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]
```

```
>sum(1:2:30), sum(1/(1:2:30))
```

```
225  
2.33587263431
```

```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret secara sim
```

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

```
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{3^k + k} = \sum_{k=0}^{\infty} \frac{1}{3^k + k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{2k-1} = \sum_{k=1}^{\infty} \frac{(-1)^k}{2k-1}$$

```
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs(x)<1)
```

$$a \sum_{k=0}^{\infty} x^k = \frac{a}{1-x}$$

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi mengenai materi kalkulus.
- mencari latihan soal di buku dan internet.
- Mempelajari perintah perintah yang ada di EMT berkaitan dengan kalkulus.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam memahami perintah yang berkaitan dengan kalkulus, solusinya dengan menonton youtube dan mempelajari rumus rumus kalkulus.

>

8. Penggunaan software EMT untuk aplikasi Statistika

a. Hal hal yang dipelajari beserta contohnya

- Menyimpan data
- Menghasilkan data acak
- Membaca data yang tersimpan
- Perhitungan analisis data statistika
- Membaca grafik statistika
- Menyimpan data hasil analisis

Menyimpan Data Dalam Bentuk Matriks

Array

Array adalah kumpulan-kumpulan variabel yang menyimpan data dengan tipe yang sama atau data-data yang tersusun secara linear dimana di dalamnya terdapat elemen dengan tipe yang sama.

Vektor digunakan untuk menggambarkan array angka satu dimensi. Vektor memiliki panjang, yang merupakan jumlah elemen dalam array.

Sedangkan matriks digunakan dalam mendeskripsikan susunan bilangan dua dimensi yang disusun dalam baris dan kolom. matriks memiliki ukuran, yaitu jumlah baris dan kolom.

Hubungan antara array dan matriks adalah bahwa matriks adalah bentuk khusus dari array. Array dapat memiliki lebih dari dua dimensi, tetapi matriks selalu memiliki dua dimensi. Dalam pemrograman, array dan matriks sering digunakan untuk menyimpan data dalam jumlah besar dan memudahkan pengaksesan data tersebut.

Mari kita bahas beberapa hal terkait vektor terlebih dahulu

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

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

```
>w=intrandom(10,12)
```

```
[1, 12, 12, 1, 8, 11, 11, 8, 1, 10]
```

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)
```

```
[19, 17, 17, 11, 5, 7, 14, 6, 18, 4]
```

```
>x=[5,10,15,20]
```

```
[5, 10, 15, 20]
```

```
>find(x,s)
```

```
[3, 3, 3, 2, 1, 1, 2, 1, 3, 0]
```

Berikutnya adalah cara mencari indeks dari sebuah vektor dengan contoh vektor. Untuk indeks pada EMT berbeda dengan indeks pada Python yang kita pelajari sebelumnya di Algoritma dan pemrograman. Perbedaannya jika sebelumnya untuk menentukan indeks akan dimulai dari nol namun di EMT akan dimulai dari angka satu, berikut penjelasannya

```
>indexof(w,1:10)
```

```
[1, 0, 0, 0, 0, 0, 0, 5, 0, 10]
```

```
>x= sort(intrandom(10,12))
```

```
[3, 4, 6, 7, 8, 8, 9, 10, 11, 11]
```

```
>indexofsorted(x,1:15)
```

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

```
>z=intrandom(1000,10); multofsorted(sort(z),1:10), sum(%)
```

```
[79, 94, 101, 111, 87, 98, 107, 111, 95, 117]  
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)
```

0.31232	0.92519	0.014936	0.30395
0.41988	0.042611	0.93098	0.98727
0.2075	0.42556	0.42418	0.49088

```
>shortformat; A=intrandom(5,4,20)
```

6	2	20	2
20	12	17	3
16	8	14	10
3	5	16	14
20	14	17	11

```
>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)
```

0.45954	0.55028	0.96497
0.74057	0.95865	0.27366
0.4379	0.64177	0.93735
0.65758	0.20468	0
0.70238	0.47586	0

```
>for k=1 to prod(size(A)); A{k}=k; end; short A
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

```
>B=zeros(size(A))
```

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

```
>B=ones(size(A))
```

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Berikutnya operasi penjumlahan dan pengurangan matriks

```
>shortformat; I=intrandom(3,4,10)
```

3	6	4	10
5	10	5	2
1	2	9	7

```
>shortformat; J=intrandom(3,4,8)
```

2	6	5	7
2	7	6	2
4	1	6	6

```
>C= I-J
```

1	0	-1	3
3	3	-1	0
-3	1	3	1

```
>C= I+J
```

5	12	9	17
7	17	11	4
5	3	15	13

Dalam materi matriks yang pernah kita pelajari ada sebutan transpose, Invers dan juga determinan, jika menggunakan EMt sebagai berikut secara berurutan:

```
>T = transpose(I)
```

3	5	1
6	10	2
4	5	9
10	2	7

```
>T = I'
```

3	5	1
6	10	2
4	5	9
10	2	7

```
>K = J^(-1)
```

0.5	0.16667	0.2	0.14286
0.5	0.14286	0.16667	0.5
0.25	1	0.16667	0.16667

```
>shortformat; L=intrandom(3,3,7)
```

2	3	3
2	4	1
6	6	3

```
>det(L)
```

-24

Selanjutnya adalah cara ekstraksi baris dan kolom, atau sub-matriks,yang mirip dengan R sebagai berikut:

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

3	3
4	1
6	3

```
>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,
```

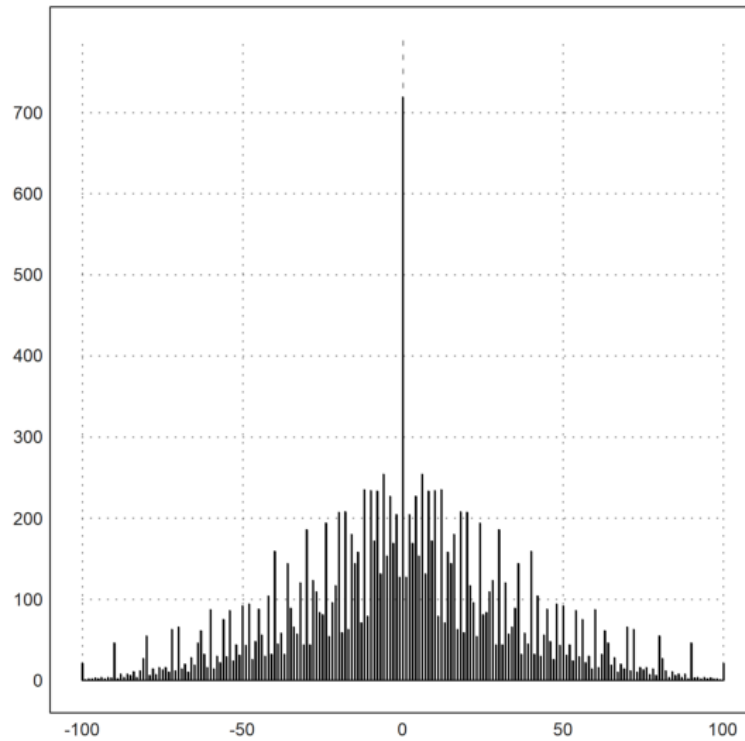
1	2	3	0	5
6	7	0	9	10

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

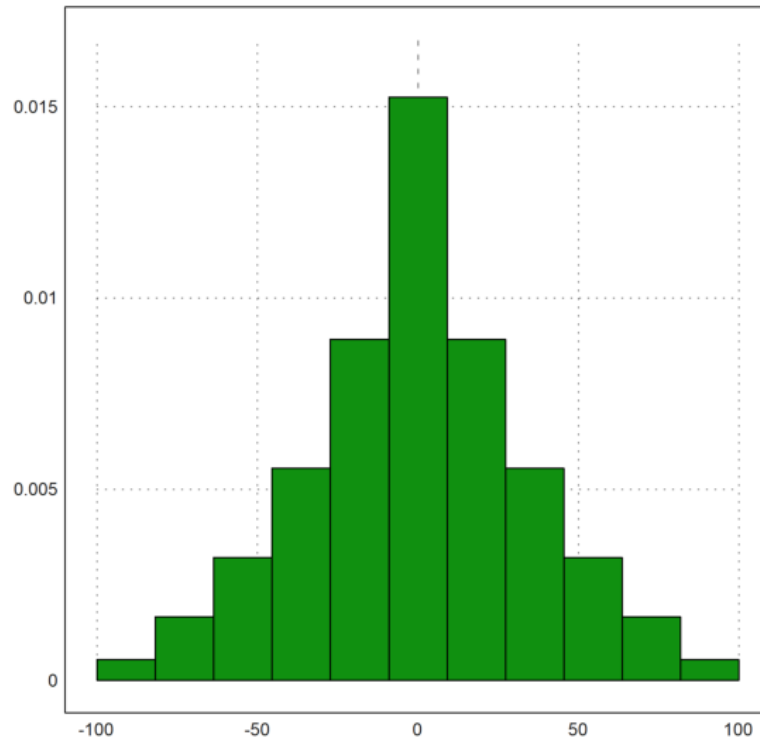
```
>a=0:10; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
>u=sort(unique(q)); f=getmultiplicities(u,q); ...
>statplot(u,f,"h"):
```



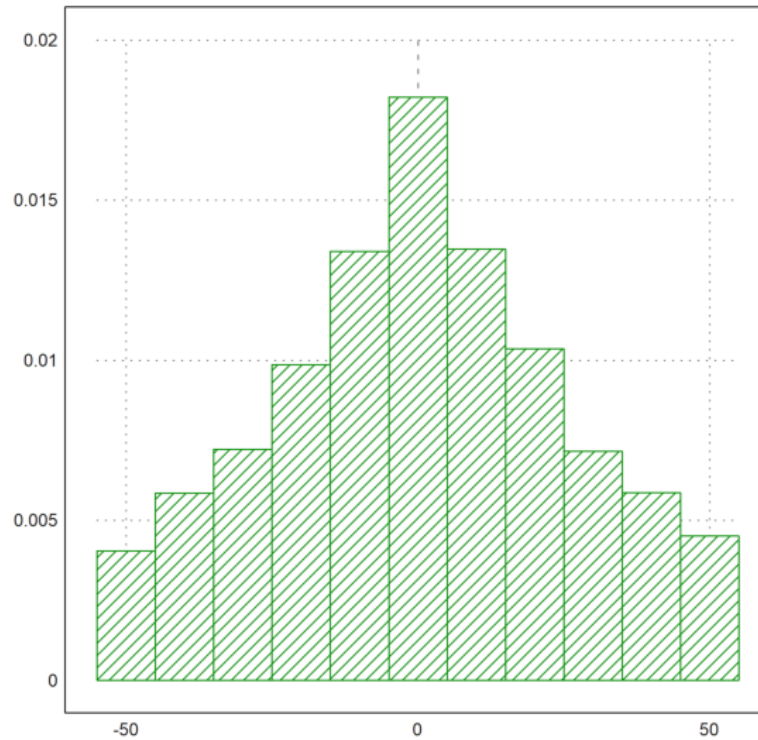
```
>getfrequencies(q,-50:10:50)
```

```
[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="/"):
```



Menghasilkan Data Acak Menggunakan Fungsi Distribusi CAKUPAN
MATERI

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)
```

```
[3, 8, 3, 2, 3, 1, 5, 1, 8, 7]
```

2. Pengertian Distribusi Diskrit dan Konsep yang Terkait

Distribusi diskrit dalam statistika adalah distribusi data yang

memiliki nilai-nilai yang terpisah dan dapat dihitung. Contohnya adalah jumlah anak dalam sebuah keluarga, jumlah mata dadu yang muncul, atau jumlah pelanggan yang datang ke sebuah toko.

Distribusi diskrit merujuk pada distribusi

probabilitas yang melibatkan variabel acak diskrit. Variabel acak diskrit adalah variabel acak yang hanya dapat mengambil nilai-nilai terpisah, bukan nilai-nilai kontinu seperti pada variabel acak kontinu. Distribusi diskrit memberikan probabilitas masing-masing nilai yang mungkin dari variabel acak tersebut. Berikut adalah beberapa konsep kunci yang terkait dengan distribusi diskrit dalam statistika:

1. Fungsi Probabilitas Diskrit (Probability Mass Function - PMF):

-Fungsi probabilitas diskrit, atau PMF, memberikan probabilitas bahwa variabel acak diskrit akan mengambil nilai tertentu.

PMF umumnya dilambangkan dengan $P(X=x)$, di mana X adalah variabel acak dan x adalah nilai yang mungkin dari variabel tersebut.

2. Ruang Sampel (Sample Space):

- Ruang sampel adalah himpunan semua hasil mungkin dari suatu percobaan acak yang dapat diukur.
- Setiap elemen dalam ruang sampel merupakan hasil yang mungkin dari variabel acak.

3. Hukum Probabilitas untuk Distribusi Diskrit:

Probabilitas suatu kejadian adalah bilangan yang berada dalam rentang 0 hingga 1, atau $0 \leq P(A) \leq 1$ untuk setiap kejadian A.
Probabilitas total dari semua hasil dalam ruang sampel adalah 1, atau $P(S) = 1$, di mana S adalah ruang sampel.

4. Fungsi Distribusi Kumulatif (Cumulative Distribution Function - CDF):

- Fungsi distribusi kumulatif memberikan probabilitas bahwa variabel acak diskrit kurang dari atau sama dengan nilai tertentu.
- Notasi matematisnya sering kali disimbolkan sebagai $F(x) = P(X \leq x)$

5. Harapan (Expectation) dan Varians:

-Harapan atau nilai rata-rata ($E(X)$) dari distribusi diskrit adalah jumlah tertimbang dari nilai-nilai mungkin berdasarkan probabilitas masing-masing nilai.

-Varians $Var(X)$) 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]
```

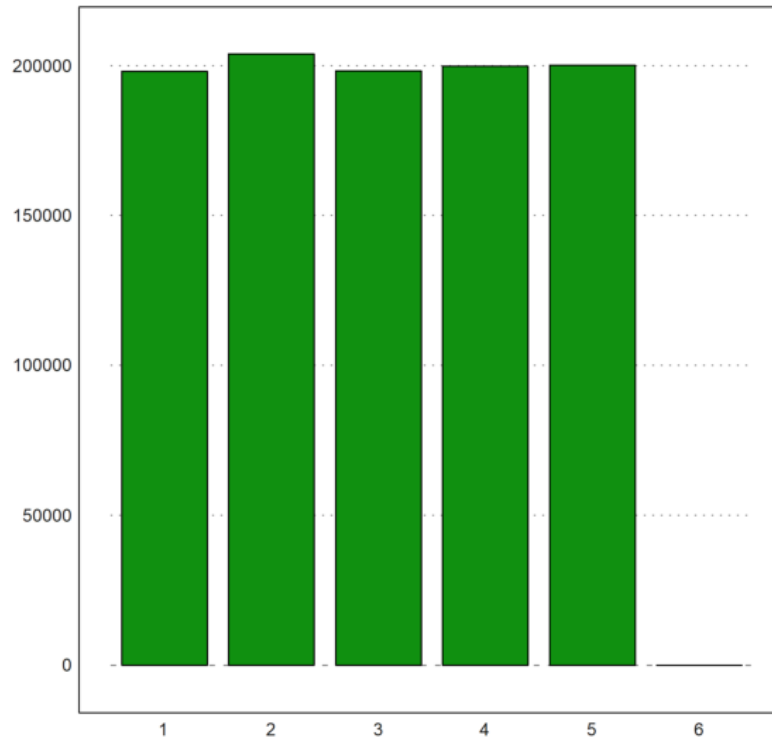
Artinya dengan probabilitas $wd[i+1]-wd[i]$ kita menghasilkan nilai acak i .

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))
```

Kita dapat menghitung distribusi binomial. Pertama ada `binomsum()`, yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

```
>bindis(410,1000,0.4)
```

```
0.7514
```

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
```

membaca data dari CSV

pertama tama kita download file csv yang telah di sediakan di besmart, setelah itu kita jadi satukan dalam 1 folder dengan file eml 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 keluarkan

writematrix digunakan untuk menuliskan matriks yang ada

lalu kita print datanya dengan

```
>printfile(file)
```

```
0.2715906836489253,0.7715320729558478,0.6110298176936501  
0.4607076408155958,0.7475173601596502,0.5399615927118507  
0.8808809313340481,0.1926857542145481,0.3004269774706536
```

titik desimal pada data tersebut dapat di jadikan pada format EMT dengan cara menggunakan readmatrix()

```
>readmatrix(file)
```

0.27159	0.77153	0.61103
0.46071	0.74752	0.53996
0.88088	0.19269	0.30043

```
>
```

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;
```

Sub Topik 5: Perhitungan terkait analisis data statistika deskriptif

Rata-rata, simpangan baku, jangkauan, modus, ukuran data, varians dan median.
Analisis data statistika deskriptif

Statistika deskriptif adalah bidang ilmu statistika yang mempelajari cara-cara untuk pengumpulan, penyusunan, dan penyajian data sehingga memberikan informasi yang berguna. Perlu diketahui juga bahwa statistika deskriptif memberikan informasi hanya mengenai data yang dipunyai dan sama sekali tidak menarik 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

$$x_1, x_2, x_3, \dots, x_n$$

adalah data yang dikumpulkan dari suatu sampel atau populasi maka rata-rata hitung untuk sampel disimbolkan dengan

$$\bar{x}$$

dan rata-rata hitung untuk populasi disimbolkan dengan

$$\mu$$

Sehingga, untuk mencari rata-rata hitung data tunggal terdapat 2 jenis rumus sebagai berikut :

1. Rata-rata hitung sampel

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

2. Rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^n X_i}{n}$$

Untuk menghitung rata-rata data tunggal dengan EMT, kita dapat menggunakan sintaks

```
> mean ([data])
```

Contoh Soal:

1. Diketahui data usia(dalam tahun) penduduk suatu daerah adalah sebagai berikut:

60,70,66,75,77,68,45,30,15,71,69,84,13

hitunglah rata-rata usia penduduk tersebut.

Jawab :

```
>mean([60,70,66,75,77,68,45,30,15,71,69,84,13])
```

57.1538461538

Jadi, rata rata data tersebut adalah 57.1538461538

2. Nilai ulangan matematika dari 10 siswa adalah 80, 88, 70, 60, 90, 75, 92, 78, 67, 90. Tentukan rata-rata dari data tersebut!

Jawab :

```
>mean([80, 88, 70, 60, 90, 75, 92, 78, 67, 90])
```

79

Jadi, rata-rata dari data tersebut yaitu 79

b. Rata-rata data tabel distribusi

Jika diberikan data

$$x_1, x_2, \dots, x_n$$

yang memiliki frekuensi berturut- turut

$$f_1, f_2, \dots, f_n$$

maka, rataaan 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 tab
```

```
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

$$t_1, t_2, \dots, t_n$$

dan masing-masing frekuensinya adalah

$$f_1, f_2, \dots, f_n$$

Untuk mencari rata rata hitung data tersebut terdapat 2 jenis rumus sebagai berikut :

1. Rata-rata hitung sampel

untuk rata-rata hitung sampel,

$$\bar{x} = \frac{\sum_{i=1}^n t_i f_i}{\sum_{i=1}^n f_i}$$

2. Rata-rata hitung populasi

untuk rata-rata hitung populasi,

$$\mu = \frac{\sum_{i=1}^n t_i f_i}{\sum_{i=1}^n f_i}$$

Untuk menghitung rata-rata data kelompok di EMT dapat dilakukan dengan langkah berikut :

1. Menentukan tepi bawah kelas(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", "tepi 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

$$x_1, x_2, \dots, x_n$$

untuk menentukan letak median dengan menggunakan rumus :

1. Jika jumlah suatu data(n) berjumlah ganjil maka nilai mediannya adalah sama dengan data yang memiliki nilai di urutan paling tengah yang memiliki nomor urut k, dimana untuk menentukan nilai k dapat dihitung menggunakan rumus:

$$k = \frac{n + 1}{2}$$

2. Jika jumlah suatu data (n) berjumlah genap, maka untuk menghitung mediannya dengan menggunakan rumus :

$$k = \frac{n}{2}$$
$$Median = \frac{1}{2}(x_k + x_{k+1})$$

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 bukan 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 (T_b), panjang kelas (P), dan tepi atas kelas (T_a) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

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 :

a. Varians data tunggal

Rumus untuk varians data tunggal berikut :

1) Untuk populasi

$$\sigma^2 = \frac{\sum_{i=1}^n (x - \mu)^2}{n}$$

2) Untuk sampel

$$S^2 = \frac{\sum_{i=1}^n (x - \bar{x})^2}{n - 1}$$

Pada EMT, untuk menemukan suatu Ragam data tunggal dapat menggunakan perintah berikut:

```
> mean(dev^2)
```

Contoh soal

Hitunglah nilai varians dari data sampel nilai siswa: 9, 10, 6, 7!

```
>data=[9, 10, 6, 7]; //mendefinisikan data  
>urut=sort(data) //mengurutkan data
```

```
[6, 7, 9, 10]
```

```
>xbar=mean(urut) //menghitung rata rata dari data
```

```
8
```

```
>dev= urut-xbar
```

```
[-2, -1, 1, 2]
```

```
>varians=mean(dev^2) //menghitung varians
```

```
2.5
```

Jadi, varians dari data sampel tersebut adalah 2.5

b. Varians data kelompok

Rumus untuk varians data kelompok sebagai berikut :

1) Untuk populasi

$$\sigma^2 = \frac{\sum_{i=1}^n f_i (x_i - \mu)^2}{\sum_{i=1}^n f_i}$$

2) Untuk sampel

$$S^2 = \frac{\sum_{i=1}^n f_i (x_i - \bar{x})^2}{\sum_{i=1}^n f_i - 1}$$

Pada EMT, untuk menemukan Ragam data berkelompok dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas (T_b), panjang kelas (P), dan tepi atas kelas (T_a) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

```
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];  
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

3. Menghitung Ragam dengan perintah

```
> (T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]);m=mean(t,f);  
> sum(f*(t-m)^2)/sum(f) //untuk populasi  
> sum(f*(t-m)^2)/(sum(f)-1) //untuk sampel
```

Contoh soal

Tentukan varians data sampel dari tabel berikut :

```
>printfile("Tabel data kelompok varians.dat",7)
```

```
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/variens. 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 berkelompok dapat menggunakan perintah berikut:

1. Menentukan tepi bawah kelas (T_b), panjang kelas (P), dan tepi atas kelas (T_a) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

```
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];  
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

3. Menghitung Ragam dengan perintah

```
> (T[,1]+T[,2])/2; t=fold(r,[0.5,0.5]);m=mean(t,f);  
> 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

$$x_1, x_2, \dots, x_n$$

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 (T_b), panjang kelas (P), dan tepi atas kelas (T_a) dengan rumus :

$$T_b = a - 0,5$$

$$P = (b - a) + 1$$

$$T_a = b + 0.5$$

dengan a = batas bawah kelas dan b = batas atas kelas

2. Mendeskripsikan data dalam bentuk tabel, dengan perintah

```
> r=tepi bawah terkecil:panjang kelas:tepi atas terbesar; f=[frekuensi];
```

```
> T:=r[1:jumlah kelas]' | r[2:jumlah kelas + 1]' | f'; writetable(T,labc=["tepi bawah", "tepi atas", "frekuensi"])
```

3. Menghitung jangkauan data berkelompok

```
> max(transpose(T[,2]))-min(transpose(T[,1]))
```

Contoh soal :

Berikut adalah data hasil dari pengukuran berat badan 20 siswa SD kelas V. Dari ke 20 siswa, siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 4 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 3 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 2 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 3 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 3 orang. Tentukan jangkauan dari

data hasil pengukuran berat badan 20 siswa di SD tersebut!

Jawab :

```
> printfile("Tabel jangkauan data kelompok.dat",7) //menyederhanakan informasi
```

```
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 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 pembagiannya

Contoh soal

1. Tentukan Q1,Q2 dan Q3 dari data : 7,3,8,5,9,4,8,3,10,2,7,6,8,7,2,6,9.

```
>data=[7,3,8,5,9,4,8,3,10,2,7,6,8,7,2,6,9];  
>urut=sort(data)
```

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

```
>quartiles(urut)
```

```
[2, 3.5, 7, 8, 10]
```

dari hasil di atas diperoleh nilai sebagai berikut :

Nilai minimal data = 2

Q1=3.5

Q2=7

Q3=8

Nilai maksimal data = 10

2. Tentukan D8 dari data : 6,3,8,9,5,9,9,7,5,7,4,5,8,3,7,6

```
>data=[6,3,8,9,5,9,9,7,5,7,4,5,8,3,7,6];  
>urut=sort(data)
```

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

```
>quantile(urut,0.8) //nilai 0.8 didapatkan 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

Sub Topik 6: Menggambar Grafik Statistika

Diagram Kotak

Diagram kotak atau box plot merupakan ringkasan distribusi sampel yang disajikan secara grafis yang bisa menggambarkan bentuk distribusi data (skewness), ukuran tendensi sentral dan ukuran penyebaran (keragaman) data pengamatan. Diagram kotak sering digunakan ketika jumlah distribusi data perlu dibandingkan. Diagram kotak menyajikan informasi tentang nilai-nilai inti dalam distribusi data termasuk juga pencilan. Pencilan adalah titik data yang terpaut jauh dari titik data lainnya.

Contoh:

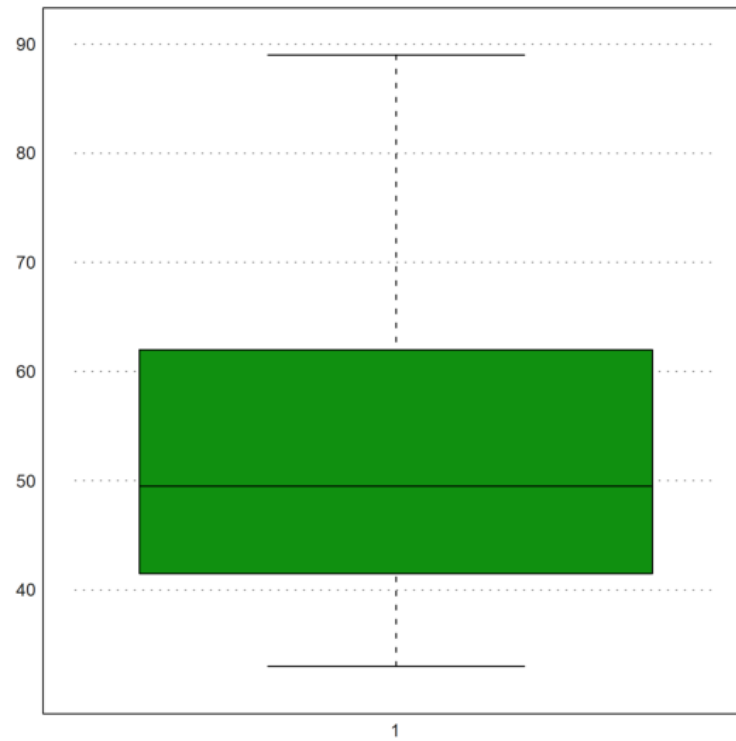
Diketahui data berat badan mahasiswa di Universitas A sebagai berikut.

```
>A=[55,50,33,42,44,37,63,74,56,34,51,43,45,39,64,77,60,35,53,43,48,41,65,87,61,36,54,44,49,41,66,89]
```

```
[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.

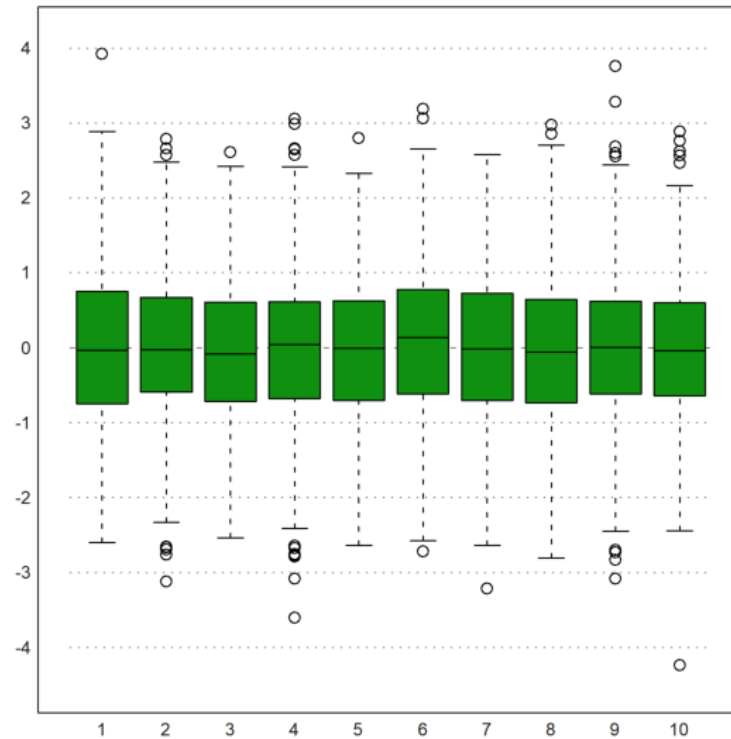
```
>boxplot(A):
```



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

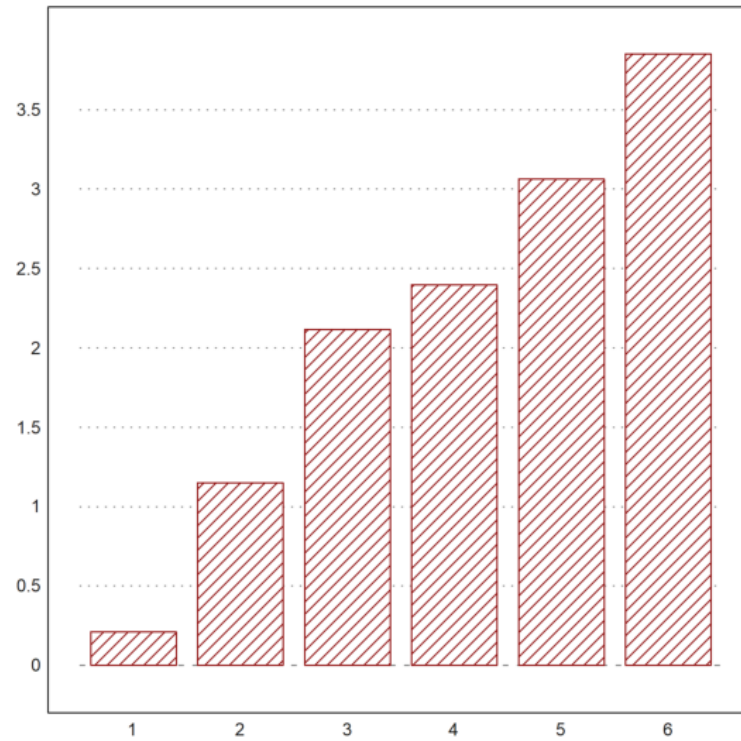
Diagram Batang

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.

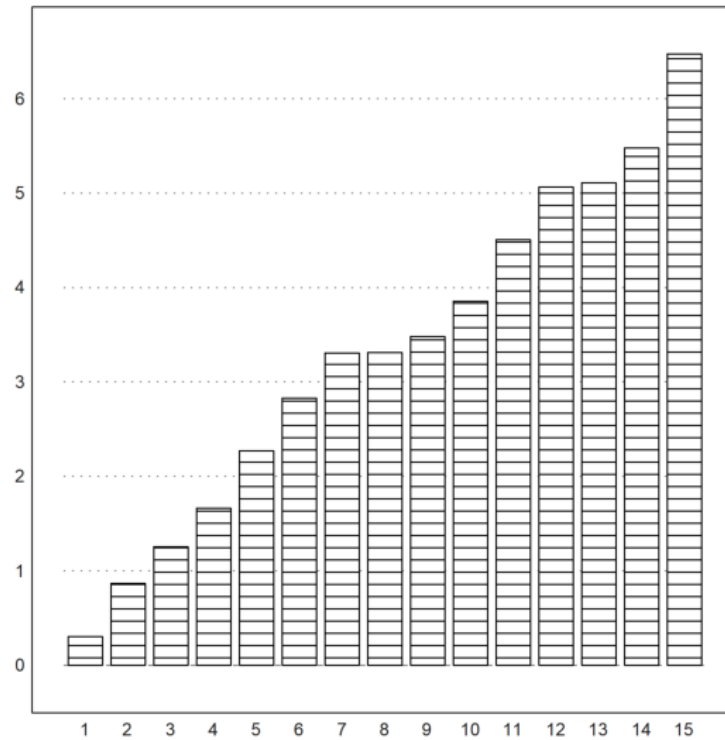
Contoh:

Kita akan membuat diagram batang secara random.

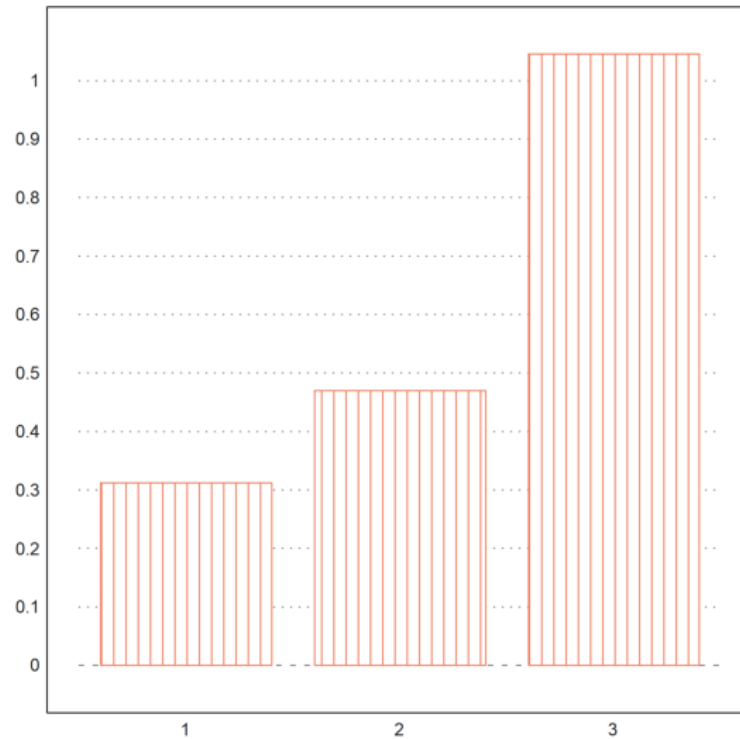
```
>columnplot(cumsum(random(6)),style="/",color=red):
```



```
>columnspot(cumsum(random(15)),style="-",color=black):
```

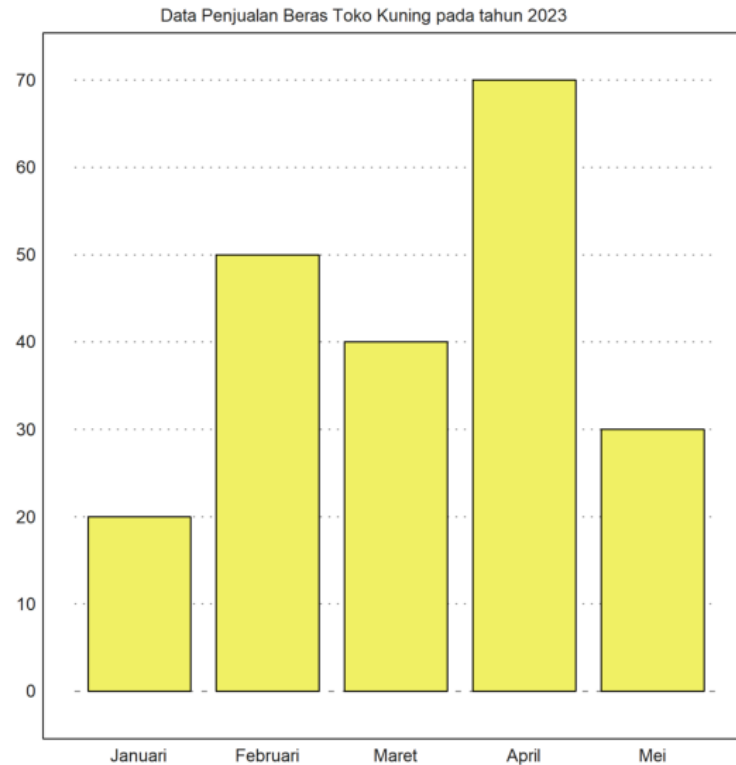


```
>columnplot(cumsum(random(3)),style="|",color=orange):
```



Selanjutnya kita akan mencoba membuat diagram batang penjualan yang menggunakan variabel.

```
>months=["Januari","Februari","Maret","April","Mei"];  
>values=[20,50,40,70,30];  
>columnplot(values,lab=months,color=yellow);  
>title("Data Penjualan Beras Toko Kuning pada tahun 2023"):
```

Perintah `"columnplot(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.

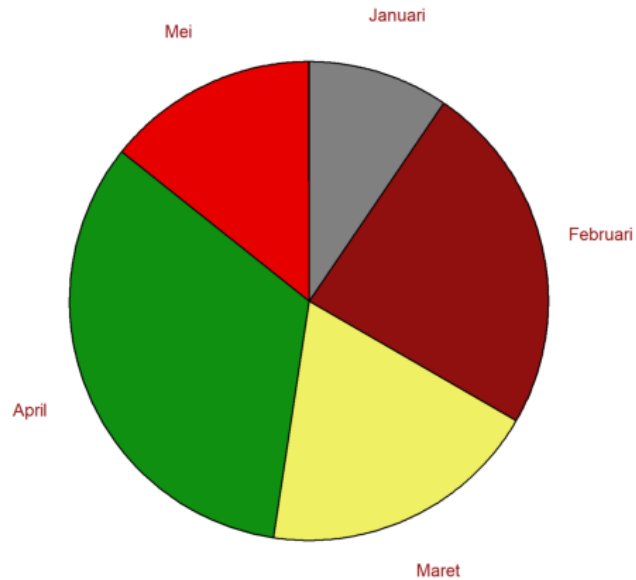
Diagram Lingkaran

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]
```

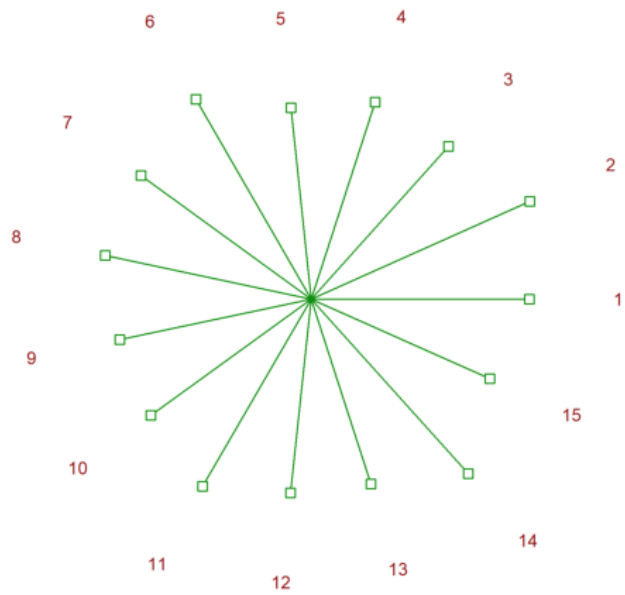
```
>i=[1,2,3,4,5]; piechart(values[i],color=CP[i],lab=months[i]):
```



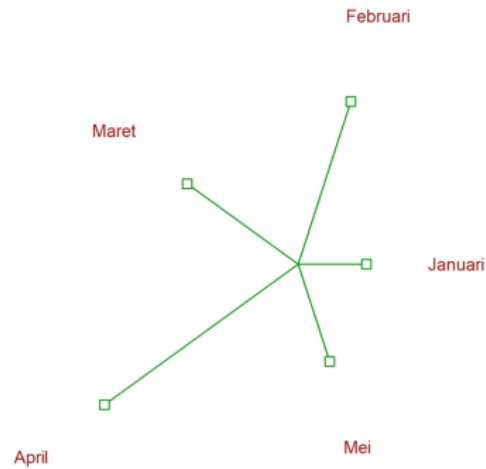
RGB adalah singkatan dari Red, Green, and Blue, dan setiap parameter mendefinisikan intensitas warna dengan nilai antara 0 dan 1. Warna pertama dalam daftar adalah warna abu-abu dengan jumlah merah, hijau, dan biru yang sama. Warna kedua merah, ketiga kuning, dan keempat hijau. Warna terakhir adalah warna merah dengan lebih banyak merah daripada hijau atau biru.

Diagram bintang, 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):
```

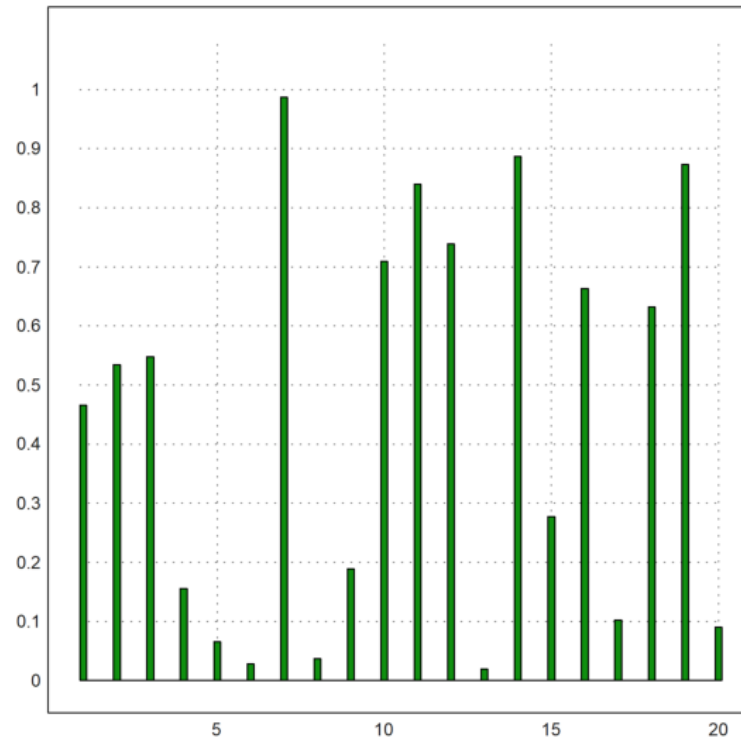


Syntax `starplot(values,lab=months,rays)` adalah perintah untuk membuat grafik bintang (star plot) dengan menggunakan nilai-nilai yang diberikan dalam vektor `values`, label sumbu yang diberikan dalam vektor `months`, dan jumlah `rays` yang menentukan jumlah garis radial yang digunakan dalam grafik

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.

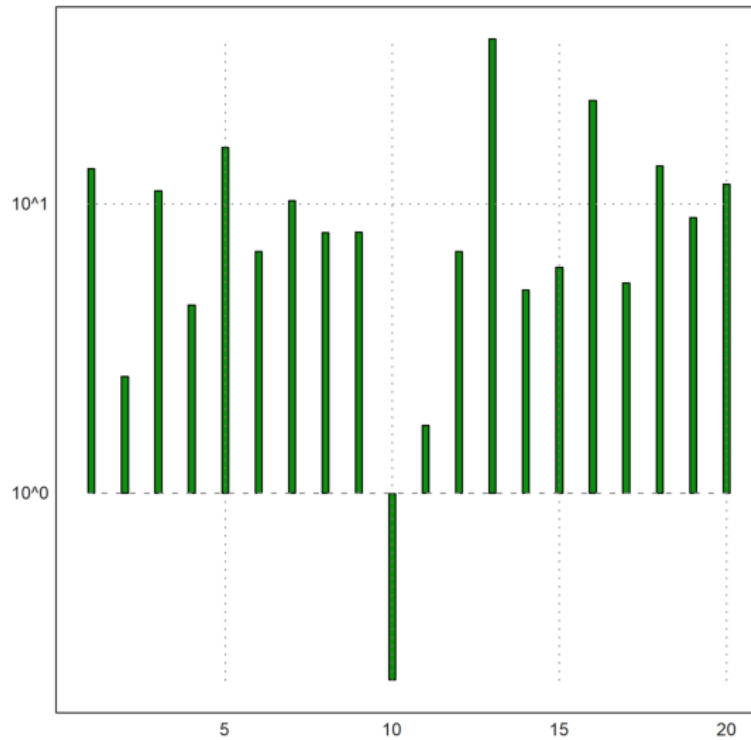
Berikut adalah plot impuls dari data acak 1 sampai 20, terdistribusi secara merata di $[0,1]$.

```
>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):
```

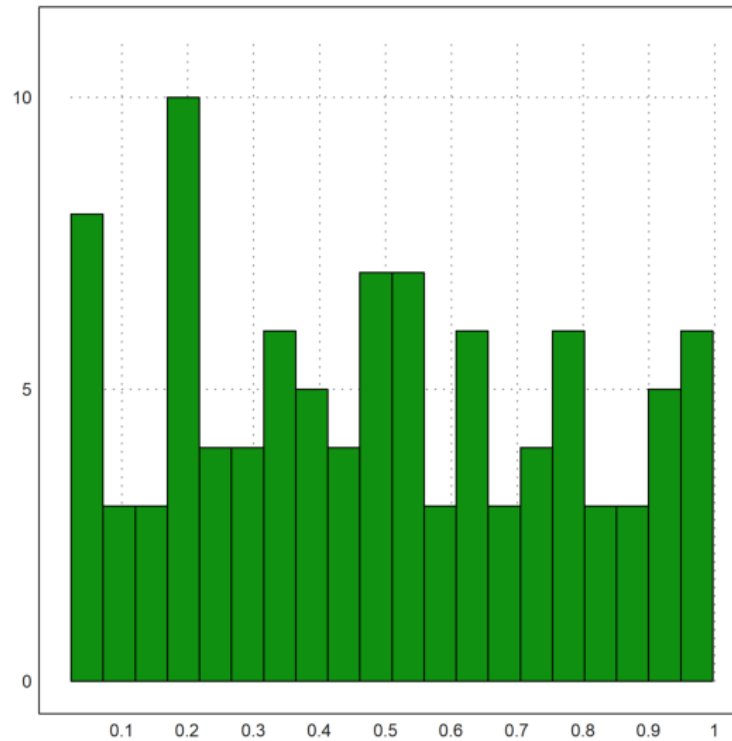



Jadi gambar grafiknya terlihat naik turun (mengalami perubahan).

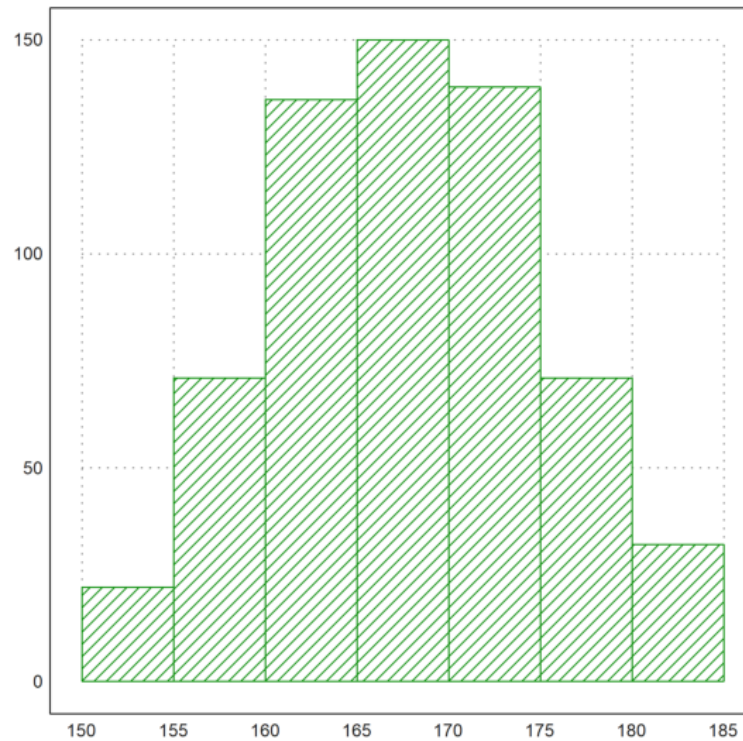
Histogram

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="/"):
```



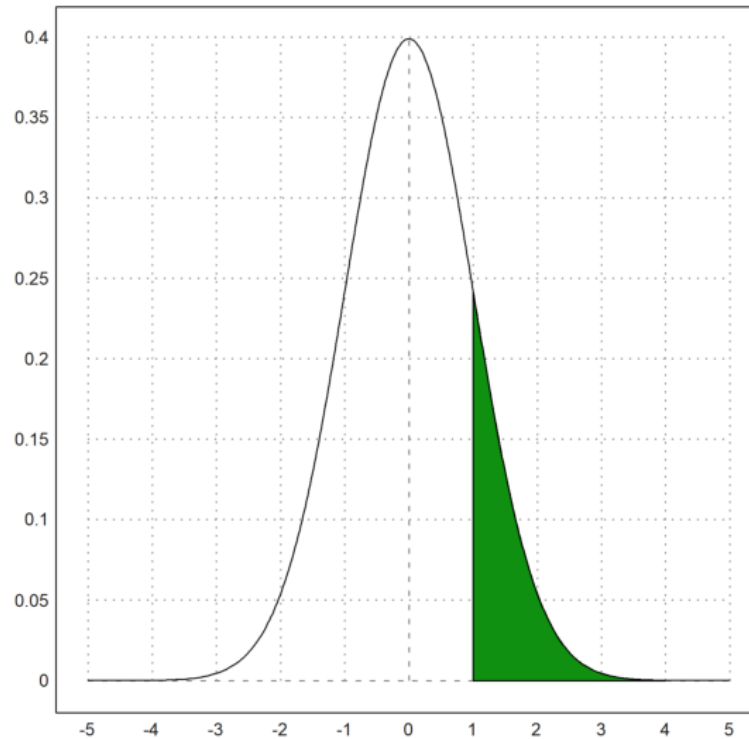
Pola "r=150:5:185" berarti bahwa nilai r dimulai dari 150, kemudian bertambah 5 setiap kali, dan berakhir saat mencapai atau melebihi 185. Dengan pola ini, kita dapat menentukan nilai-nilai r yang sesuai.

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
 $P(1 < X < 4) = \text{luas daerah hijau}$

Kurva Fungsi Distribusi Kumulatif

Cumulative Distribution Function (CDF) atau fungsi distribusi kumulatif adalah fungsi matematika yang digunakan untuk menghitung probabilitas variabel acak diskrit atau kontinu. CDF memberikan probabilitas bahwa variabel acak akan menghasilkan nilai kurang dari atau sama dengan nilai tertentu. Dalam hal ini, CDF dapat digunakan untuk menghitung probabilitas kumulatif dari variabel acak.

Berikut merupakan contoh kurva fungsi distribusi kumulatif kontinu:

```
>plot2d("normaldis",-3,5):
```

```
Function plot2d not found.  
Try list ... to find functions!  
Error in:  
plot2d("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.

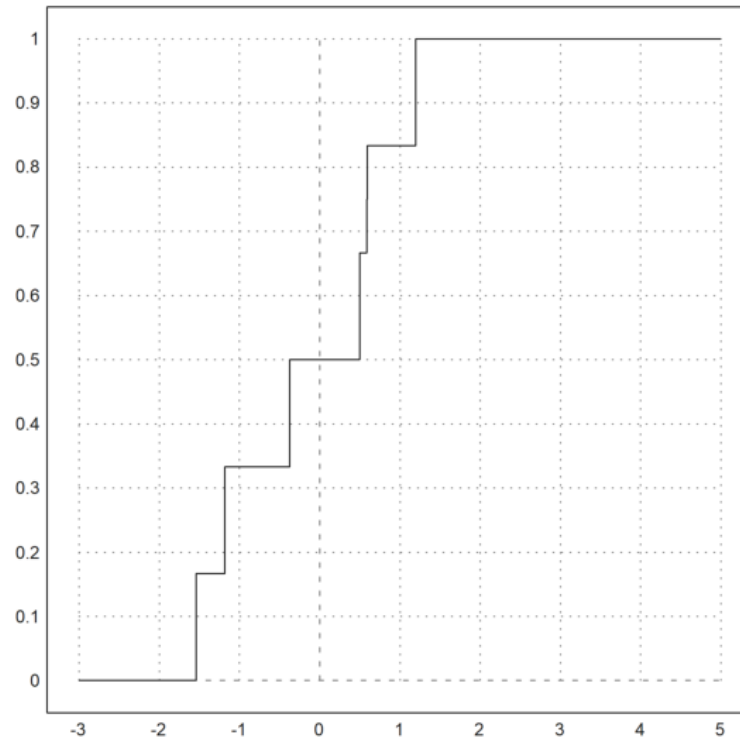
Adapun contoh kurva fungsi distribusi kumulatif diskrit sebagai berikut.

```
>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.

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";
```

Setelah 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  
0.4740947109172355,0.6328068695909269,0.5833355628808601  
0.5532677420544243,0.08495847192661377,0.9098541742831491  
0.9660500335107803,0.4155435349425303,0.9654721548152776
```

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)
```

```
0.1409789519306323,0.6345761036216137,0.3852004398321907  
0.1785920128134643,0.6613914250481273,0.04869641928896105  
0.9168512648587849,0.04738687026681564,0.3943845525023932
```

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)
```

0.140979	0.634576	0.3852
0.178592	0.661391	0.0486964
0.916851	0.0473869	0.394385

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi mengenai materi statistika.
- mencari latihan soal di buku dan internet.
- Mempelajari perintah perintah yang ada di EMT berkaitan dengan statistika.
- Mencari data untuk percobaan perintah EMT.

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam memahami perintah yang berkaitan dengan statistika, solusinya dengan menonton youtube.
- Kesulitan dalam memahami hasil perintah EMT, solusinya dengan mempelajari referensi EMT

9. Pengolahan dokumen menggunakan LaTeX.

a. Hal hal yang dipelajari beserta contohnya

- Menggunakan software LaTeX
- Menghasilkan dokumen menggunakan LaTeX
- Menggunakan Overleaf
- Membuat akun Github

Contoh gambar Overleaf dan Github

image: G2.png

image: G1.png

b. Hal hal yang dilakukan dalam mempelajari materi

- Mencari informasi mengenai LaTeX
- Mencari informasi cara membuat akun Github
- Menonton youtube dalam menggunakan Overleaf

c. Kendala kendala dan usaha untuk mengatasi kendala tersebut

- Kesulitan dalam menggunakan overleaf, solusinya dengan menonton youtube cara kerja overleaf.
-