
TUGAS PROYEK APLIKOM —

NAMA : EFINDA MAULIA AWP
KELAS : MATEMATIKA E
NIM : 23030630089

DAFTAR ISI MATERI

CAKUPAN MATERI APLIKASI KOMPUTER :

- (BAB 1) PENGENALAN SOFTWARE APLIKASI MATEMATIKA
- (BAB 2) PENGENALAN SOFTWARE EULER MATHS TOOLBOX (EMT)
- (BAB 3) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI ALJABAR
- (BAB 4) PENGGUNAAN SOFTWARE EMT UNTUK MENGGAMBAR GRAFIK 2 DIMENSI (2D)
- (BAB 5) PENGGUNAAN SOFTWARE EMT UNTUK MENGGAMBAR GRAFIK 3 DIMENSI (3D)
- (BAB 6) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI GEOMETRI
- (BAB 7) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI KALKULUS
- (BAB 8) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI STATISTIKA
- (BAB 9) PENGOLAHAN DOKUMEN MENGGUNAKAN LATEX

(BAB 1)PENGENALAN SOFTWARE APLIKASI

MATEMATIKA

Pada materi ini saya telah mempelajari berbagai macam Software maupun Aplikasi yang dapat digunakan dan dimanfaatkan untuk menyelesaikan permasalahan Matematika dengan berbagai kategorinya sesuai aplikasinya, beberapa software tersebut yakni sebagai berikut :

1. MATLAB

MATLAB (singkatan dari "matrix laboratory") adalah aplikasi untuk komputasi numerik multi-paradigma 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. Meskipun MATLAB ditujukan terutama untuk komputasi numerik, dia menyediakan fasilitas untuk komputasi simbolik menggunakan MuPAD. Paket tambahan lain, Simulink, menambahkan simulasi multi-domain grafis dan desain berbasis model untuk sistem dinamis. Pada tahun 2020, MATLAB memiliki lebih dari 4 juta pengguna di seluruh dunia. Pengguna MATLAB berasal dari berbagai latar belakang teknik, sains, dan ekonomi.

2. Graphmatica

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

grafik fungsi Cartesian, relasi, dan pertidaksamaan, persamaan diferensial polar, parametrik, dan biasa. hingga 999 grafik di layar sekaligus. plot dan kurva data baru fitur pencocokan kurva memecahkan secara numerik dan menampilkan garis singgung dan integral secara grafis. menemukan titik kritis, solusi persamaan, dan titik potong kurva. mencetak grafik Anda, salin ke clipboard sebagai bitmap atau metafile yang disempurnakan dalam warna hitam dan putih atau berwarna, atau ekspor ke file JPEG / PNG.

3. 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.

4. GNU Octave

GNU Octave adalah bahasa tingkat tinggi, terutama ditujukan untuk komputasi numerik. 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. Ini juga dapat digunakan sebagai bahasa berorientasi batch. Oktave memiliki perintah yang lengkap untuk memecahkan masalah umum aljabar linear numerik, menemukan akar persamaan nonlinier, mengintegrasikan fungsi biasa, memanipulasi polinomial, dan mengintegrasikan persamaan diferensial dan diferensial-aljabar biasa. Ini mudah dikembangkan dan disesuaikan melalui fungsi buatan pengguna yang ditulis dalam bahasa Octave sendiri, atau menggunakan modul yang dimuat secara dinamis yang ditulis dalam C++, C, Fortran, atau bahasa lain. GNU Octave juga merupakan perangkat lunak yang dapat didistribusikan ulang secara bebas.

5. Maple

Maple dikembangkan pertama kali pada tahun 1980 oleh Grup Symbolic Computation di University of Waterloo Ontario, Kanada. Maple merupakan Computer Algebra System (CAS) yang dapat memanipulasi pola, prosedur, dan perhitungan algoritma, baik untuk analisis maupun sintesis. Maple mencakup beberapa bidang komputasi teknis, seperti matematika simbolik, analisis numerik, pemrosesan data, visualisasi, dan lainnya. Hasil perhitungan Maple bisa menjadi solusi matematika dengan metode numerik dan simbolik. Di dalam Maple terdapat simbol, sintak, dan semantik mirip seperti bahasa pemrograman. Menurut Marjuni (2007:6), Maple merupakan software matematika buatan Waterloo Maple Inc. Dengan kemampuan kerja yang cukup handal untuk menangani berbagai komputasi analitis dan numerik. Menurut Kartono (2005: 9), Salah satu kegunaan Maple ialah Maple mampu menggambarkan suatu fungsi satu dimensi, dua dimensi, atau tiga dimensi dengan beberapa fasilitas operasi yang lain. Kapasitas Maple (software) untuk komputasi simbolik mencakup tujuan umum dari sistem komputer aljabar. Misalnya, ia dapat memanipulasi ekspresi matematika dan menemukan solusi simbolis untuk masalah tertentu, seperti yang timbul dari persamaan.

6. Magma

Magma adalah suatu perangkat lunak komputer untuk penyelesaian masalah aljabar, kombinatorik, geometri, dan teori bilangan. Magma dapat dijalankan pada sistem operasi Linux dan Windows. Magma juga mendukung sejumlah database yang dirancang untuk membantu penelitian komputasi di bidang matematika yang bersifat aljabar. Magma didistribusikan oleh Computational Algebra Group di University of Sydney. Perkembangannya sangat diuntungkan dari kontribusi yang dibuat oleh banyak anggota komunitas matematika. Microsoft Mathematics
Microsoft Mathematics adalah perangkat lunak yang dirancang untuk membantu pengguna dalam mema-

hami dan memecahkan masalah matematika. Kegunaan microsoft matemathics antara lain sebagai berikut :

Pemecah masalah matematika, Visualisasi grafis, Kalkulator multifungsi, dan Pembelajaran

7. MathType

MathType adalah perangkat lunak yang digunakan untuk membuat rumus matematika dan simbol-simbol matematika yang kompleks dalam dokumen. Ini membantu pengguna dalam menyisipkan notasi matematikayangtepat dan tampilan profesional ke dalam berbagai jenis dokumen. Berikut adalah beberapa kegunaan utama dari MathType:

Pemformatan dokumen, Notasi matematika, Editor, Integrasi dengan aplikasi, Grafis, dan Presentasi

8. Derive

Derive merupakan software matematika yang digunakan untuk melakukan berbagai perhitungan aljabar, kalkulus, dan analisis matematika. Ini membantu pengguna dalam memahami, menganalisis, dan menghitung masalahmatematika yang kompleks. Berikut adalah beberapa kegunaan utama dari Derive:

Perhitungan aljabar, Kalkulus, Analisi numerik, Visualisasi grafis, Pemodelan matematika, dan Pembelajaran.

9. Wolfram Alpha

Wolfram Alpha adalah mesin pengetahuan komputasional yang memberikan jawaban dan solusi berdasarkan pertanyaan matematika dan berbagai topik lainnya. Desmos

Desmos merupakan software aplikasi matematika yang berfungsi untuk membuat, memvisualisasikan, dan memanipulasi grafik matematika dan fungs. Ini terutama digunakan dalam konteks pendidikan dan pembelajaranmatematika. Berikut adalah beberapa kegunaan utama dari Desmos :

Visualisasi fungsi matematika, Manipulasi grafik, Pembelajaran, dan Statistik

10. R

R merupakan software aplikasi matematika yang fungsi utamanya untuk analisis data. Dengan R kita dapat mendapatkan grafik suatu data dengan cepat. R digunakan untuk melakukan analisis data, visualisasi, pemodelanstatistik, dan banyak lagi. Berikut adalah beberapa kegunaan utama dari R:

Analisis data, Membuat grafik, Pemodelan statistik, Visualisasi data, Machine learning, dan Pengolahan gambar.

11. SimPy

Simpy merupakan suatu perpustakaan (library) python yang digunakan untuk komputasi matematika simbolik. Berikut adalah beberapa kegunaan SimPy :

Perhitungan simbolik, Aljabar dan kalkulus, Pembuatan fungsi dan ekspresi, Sistem persamaan, Visualisasi, Pemrosesan rumus, dan Pembelajaran

12. Scilab

Scilab merupakan perangkat lunak dengan penggunaan open source yang digunakan untuk komputasi numerik, pemodelan, analisis data, dan visualisasi. Berikut adalah beberapa kegunaan utama dari Scilab :

Komputasi numerik, Analisi data, Pemodelan, Visualisasi, Pengolahan sinyal, Pemodelan matematika, dan Pembelajaran

13. SAGE

SAGE (System for Algebra and Geometry Experimentation) merupakan software matematika yang berguna untuk melakukan berbagai jenis komputasi matematika, termasuk aljabar, geometri, analisis, dan pemodelan. Sage memiliki aplikasi luas dalam pendidikan, riset, dan analisis matematika. Bahasa pemrograman yang digunakan yaitu Python, cython. Terdapat banyak paket opensourcematematika berkualitas tinggi tercakup dalam SageMath yang merupakan berbagai libraries dari proyek open source lain. SageMath dapat digunakan untuk pengkajian kalkulus, teori bilangan mulai dari yang elementer hingga lanjut, kriptografi, aljabar komutatif, teori grup, teori graf, dan aljabar linear baik yang eksak maupun numerik. Selain itu SageMath mempunyai kemampuan grafis, animasi grafis dan program interaktif yang mudah dipahami

13. Octave

Octave merupakan perangkat lunak matematika yang sering digunakan untuk komputasi numerik, analisis data, pemodelan matematika, dan visualisasi. Kegunaan octave antara lain sebagai berikut :

Komputasi numerik, Analisi data, Visualisasi data, Pemodelan matematika, Simulasi, Pemrograman, dan Pembelajaran

(b) HAL-HAL YANG LAKUKAN DALAM MEMPELAJARI MATERI

Untuk mendapatkan pemahaman mengenai berbagai jenis Software Matematika yang ada tentunya saya telah melakukan riset yang saya peroleh dengan mencari informasi atau sumber-sumber materi yang jelas dan relevan dengan materi pada (BAB 1) PENGENALAN SOFTWARE APLIKASI MATEMATIKA ini. Selain itu, dalam mengerjakan TUGAS pada materi ini dimana kami diberi pertanyaan mengenai software software yang sudah kami cantumkan. dengan pertanyaan "Apakah kami sudah memiliki pengalaman pada software tersebut?"

Untuk itu, saya telah mempelajari lebih lanjut mengenai masing-masing software yang saya cantumkan dengan menyertakan manfaat dan fitur apa saja yang terdapat dalam software tersebut.

(c) KENDALA-KENDALA YANG SAYA HADAPI

Dalam materi ini, terdapat banyak kendala mengenai banyaknya software yang masing asing atau belum saya kenal. Selain itu, kami dituntut untuk menggolongkan kategori dan menjelaskan mengenai informasi dari software tersebut serta fitur dan pengaplikasiaannya dalam Matematika.

Oleh karena itu, usaha-usaha yang saya lakukan berupa melakukan riset yang saya peroleh dengan mencari informasi atau sumber-sumber materi yang jelas dan relevan dengan materi pada (BAB 1) PENGENALAN SOFTWARE APLIKASI MATEMATIKA ini. Terlebih lagi, dengan informasi yang masih minim atau kurang membuat saya lebih berusaha untuk mencari tahu mengenai berbagai software software Matematika tersebut.

(BAB 2) PENGENALAN SOFTWARE EULER MATHS

TOOLBOX (EMT) ** (a) HAL-HAL

YANG TELAH SAYA PELAJARI

>

Dalam Materi PENGENALAN SOFTWARE EULER MATH TOOLBOX (EMT) ini, kami dituntut untuk mempelajari salah satu software Matematika yakni EMT yang dapat melakukan beberapa pengaplikasian permasalahan dan persoalan Matematika. Pada pengenalan ini, saya mempelajari dasar-dasar yang diperlukan untuk dapat mengoperasikan Software EMT ini. Beberapa contoh perintah dan Komentar dapat dilakukan pada software EMT ini, salah satu contohnya yakni sebagai berikut :

1. UNTUK MEMBERIKAN KOMENTAR

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

Judul

Sub-Judul

$$F(x) = \int_a^x f(t) dt$$
$$\frac{x^2 - 1}{x - 1} = x + 1$$
$$\int x^3 dx = \left[\frac{x^4}{4} + 2, \frac{x^4}{4} + 2 \right]$$

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

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

- membuat baris perintah

```
>//baris perintah diawali dengan >, komentar atau keterangan diawali dengan //
```

- cara penulisan sintaks perintah pada EMT

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

Pastikan untuk menggunakan titik desimal, bukan koma desimal untuk bilangan!

Gunakan * untuk perkalian dan ^ untuk eksponen (pangkat).

Seperti biasa, * dan / bersifat lebih kuat daripada + atau -.

^ mengikat lebih kuat dari *, sehingga $\pi * r^2$ merupakan rumus luas lingkaran.

Jika perlu, Anda harus menambahkan tanda kurung, seperti pada $2^{(2^3)}$.

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

```
1.25
```

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

```
0.707106781187
```

```
-1
```

```
0.5
```

```
>log10 (100)
```

```
2
```

```
>cos(90°), tan(45°)
```

0

1

- menggunakan satuan dalam EMT dan mengubah sistem satuan menjadi standart Internasional

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
```

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

```
1609.344
```

- cara menampilkan 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 mengatrtamilan 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
```

- cara memberikan 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.

- cara menampilkan daftar variabel

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

```
>listvar
```

| | |
|---|------|
| r | 1.25 |
|---|------|

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 |

- cara menampilkan panduan pada EMT

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.

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

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

```
>random(10)
```

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

```
>normal(10)
```

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


- cara membuat matriks dan vektor pada EMT

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
```

```
>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]
```

- cara menyelesaikan bilangan kompleks pada EMT

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 kompleks 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

- cara menampilkan matematika simbolik pada EMT

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

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

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

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

$$(b + a)^2$$

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

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

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

```
3628800
```

- cara menampilkan simbolik matematika dengan Latex pada EMT
Untuk melakukan hal ini, tambahkan tanda dolar (\$) di depan tanda & pada setiap perintah Maxima.

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

$$(b + a)^2$$

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

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

(BAB 3) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI ALJABAR

Dalam materi aplikasi penggunaan software EMT ini untuk pertama kalinya, kami dikenalkan dengan PENGGUNAAN EMT UNTUK ALJABAR. Dimana perhitungan aljabar dapat diselesaikan menggunakan perintah di EMT yang tentunya memudahkan kita dalam menyelesaikan suatu permasalahan matematika. Beberapa BASIS BASIS PERINTAH yang utama dalam menyelesaikan perintah aljabar, diantaranya adalah sebagai berikut :

1. SINTAKS SINTAKS DASAR DALAM ALJABAR

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

```
30.65625
```

- Cara membuat string dalam EMT
Contoh:

```
>"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.
```

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

```
affe  
charlie  
bravo
```

- Cara menampilkan nilai boolean dalam EMT
Contoh:

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

```
0  
1
```

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

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

- Cara menampilkan format keluaran dalam EMT
Contoh:

```
>defformat; pi
```

```
3.14159265359
```

```
>printhex(pi)
```

3.243F6A8885A30*16⁰

- Cara membuat ekspresi dalam EMT
Contoh:

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

12.56637061435917

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

13

Cara membuat fungsi dalam EMT

Contoh:

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

4.472135955

Cara membuat parameter default dalam EMT

Contoh:

```
>function f(x,a=1) := a*x^2
>f(4)
```

16

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

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


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

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

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

- Cara memecahkan ekspresi
Contoh:

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

1.41421356237

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

$$\left[\left[x = \frac{bf - ce}{bd - ae}, y = \frac{cd - af}{bd - ae} \right] \right]$$

- Cara menyelesaikan pertidaksamaan
Contoh:

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

```
C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/fo\
urier_elim/fourier_elim.lisp
```

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

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

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

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

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

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

- Cara menggunakan bahasa matriks
Contoh:

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

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |

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

| |
|---|
| 3 |
| 4 |

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

| |
|--------|
| [3, 4] |
|--------|

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

| | |
|-----|------|
| -2 | 1 |
| 1.5 | -0.5 |

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

```
11  
25
```

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

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

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

```
0.493453    0.601344    0.659461    0.967468    1  
0.193151    0.935921    0.0728753    0.988966    2  
0.0104376    0.356626    0.52143     0.428893    3
```

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

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

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

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

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

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

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

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

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

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

```
>A[2]
```

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

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

```
2
```

- Cara menyortir dan mengacak
Contoh:

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

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

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

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

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

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

```
>ind
```

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

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

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

- Cara menyelesaikan aljabar linier dalam EMT
Contoh:

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

```
-4  
4.5
```

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

```
2.509104035652854e-13
```

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

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

```
>det(A)
```

0

- Cara membuat matriks simbolik
Contoh:

```
>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(%)
```

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


```
>$\text{invert}(A) \text{ with } a=0
```

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

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

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

- Cara mencari nilai numerik dalam ekspresi simbolik
Contoh:

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

$$\begin{pmatrix} 1 & 3.14159 \\ 4 & 5 \end{pmatrix}$$

```
>$&A
```

$$\begin{pmatrix} 1 & \frac{1146408}{364913} \\ 4 & 5 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 3.141592653589793 \\ 4 & 5 \end{pmatrix}$$

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

1.414213562373095

- Cara mencari demo-suku bunga
Contoh:

```
>K=5000
```

5000

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

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaks berikut.

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

```
5150
```

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

$$R + q K$$

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

$$q^3 R + q^2 R + q R + R + q^4 K$$

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

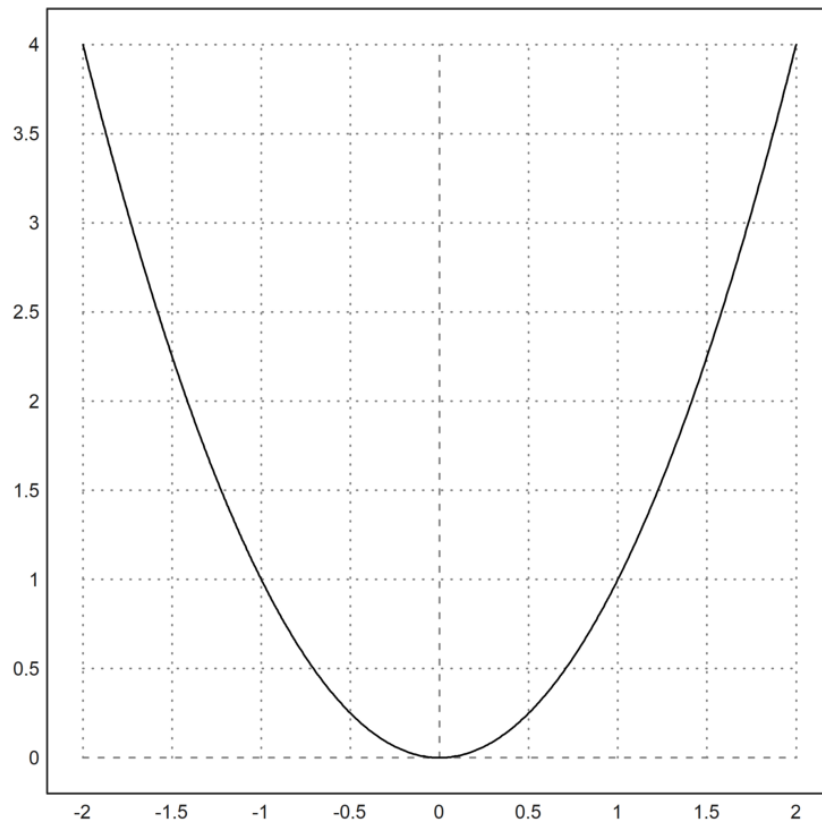
$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}$$

(BAB 4) PENGGUNAAN SOFTWARE EMT UNTUK

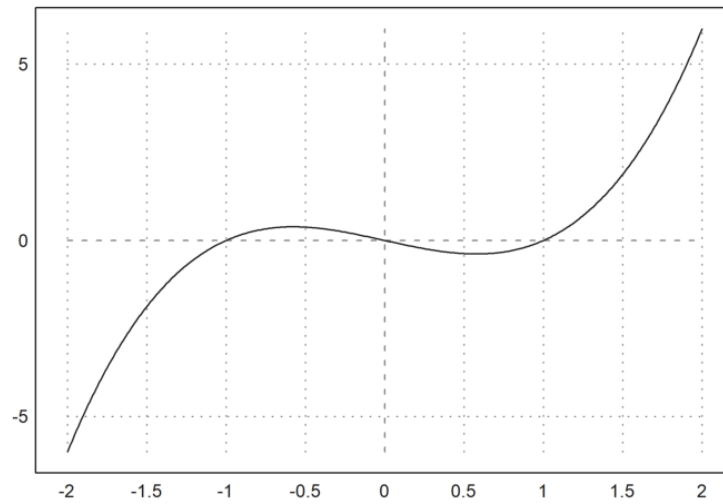
MENGGAMBAR GRAFIK 2 DIMENSI

Dalam materi ini, kami diperkenalkan dengan cara menggambar grafik pada bidang KARTESIUS atau BIDANG-XY atau DALAM 2 DIMENSI, dimana dengan menggunakan perintah syntax PLOT2D maka kita dapat menghasilkan gambar dari grafik permasalahan atau soal MATEMATIKA yang kita tanyakan. Beberapa syntax dan perintah khusus diperlukan untuk menjalankan PLOT2D yakni diantaranya adalah sebagai berikut :

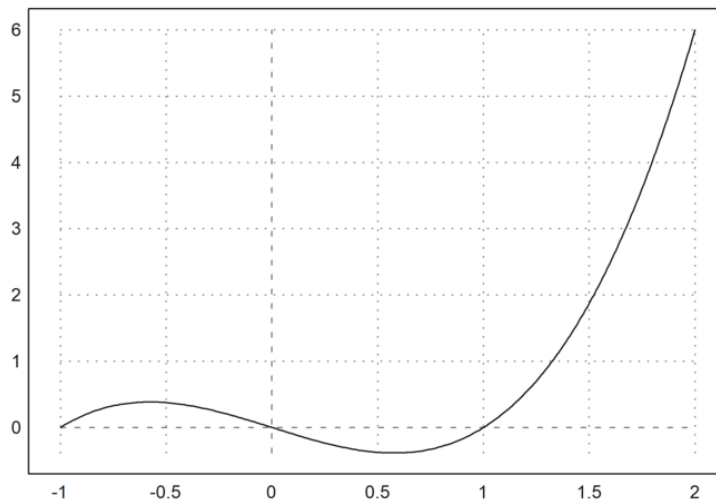
```
>plot2d("x^2"):
```



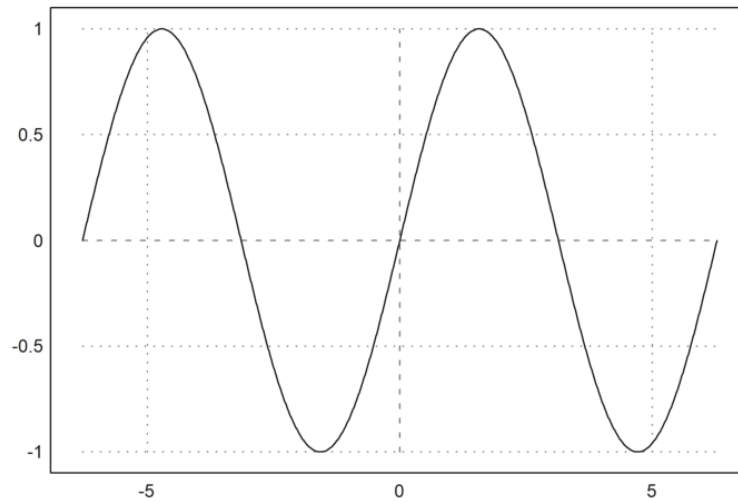
```
>aspect(1.5); plot2d("x^3-x"):
```



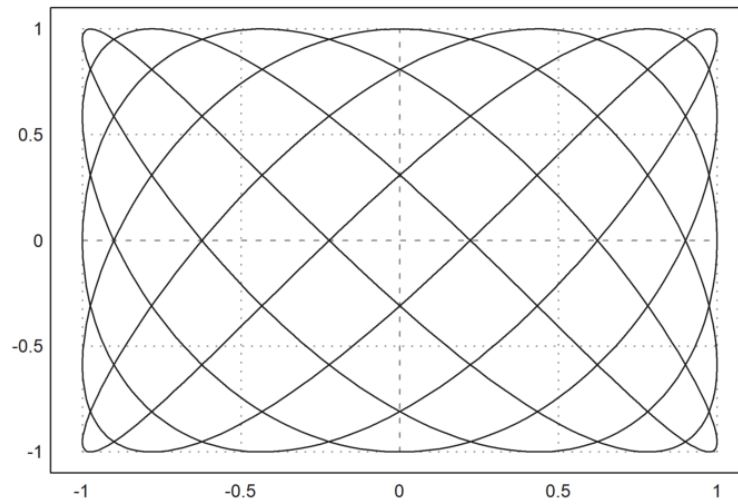
```
>plot2d("x^3-x",-1,2):
```



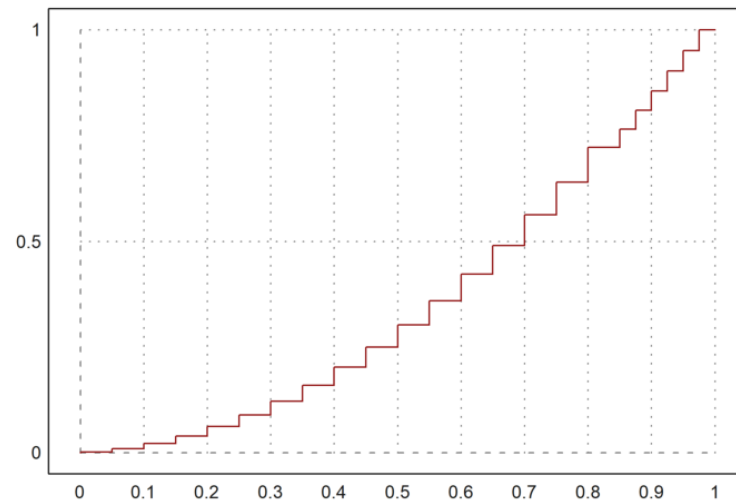
```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval  $[-2\pi, 2\pi]$ 
```

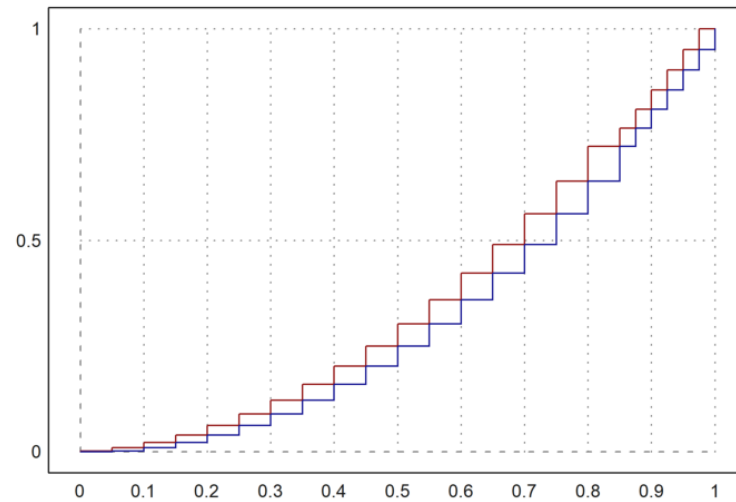
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



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

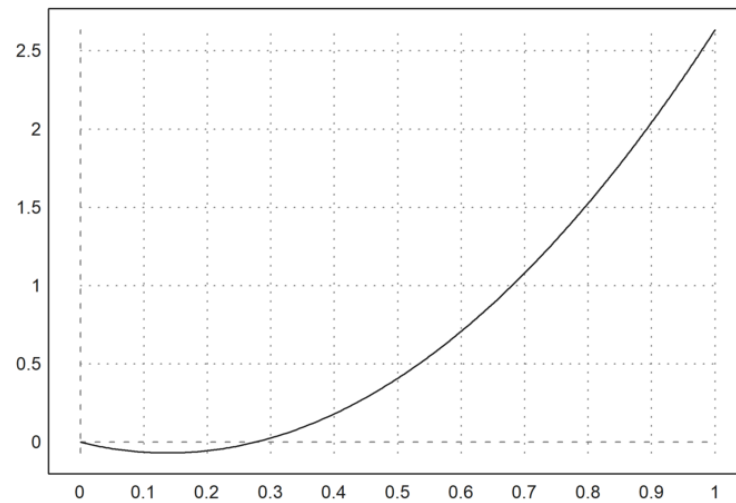


```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

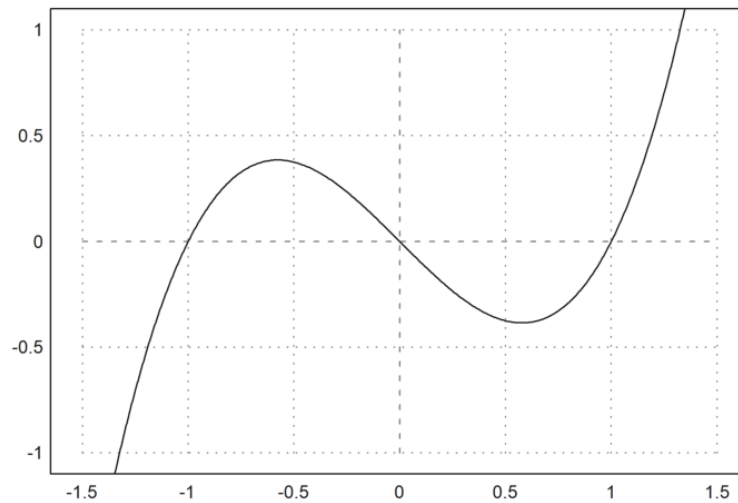


- Cara membuat grafik fungsi dalam satu parameter
Contoh:

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



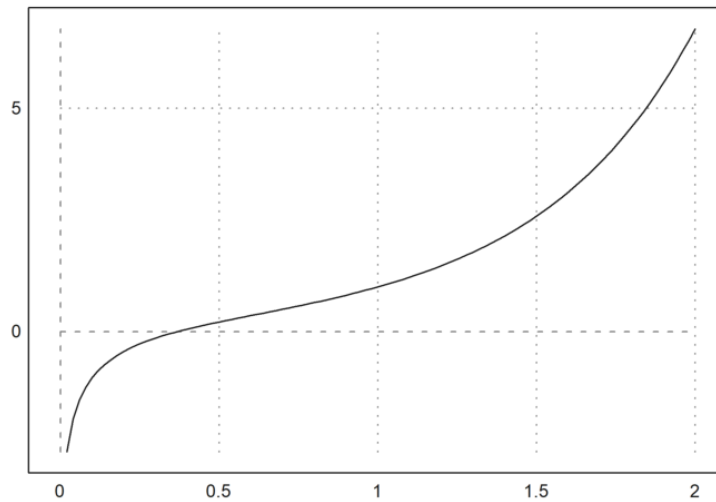
```
>function f(x) := x^3-x; ...  
>plot2d("f",r=1):
```



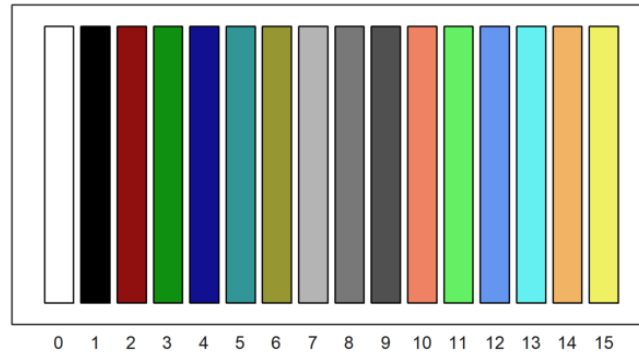
```
>function f(x) &= diff(x^x,x); $f(x)
```

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

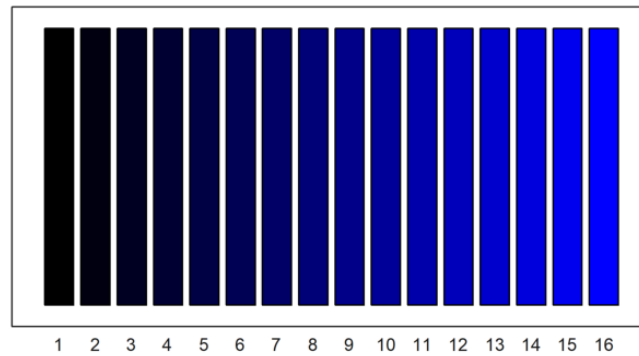
```
>plot2d(f,0,2):
```



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

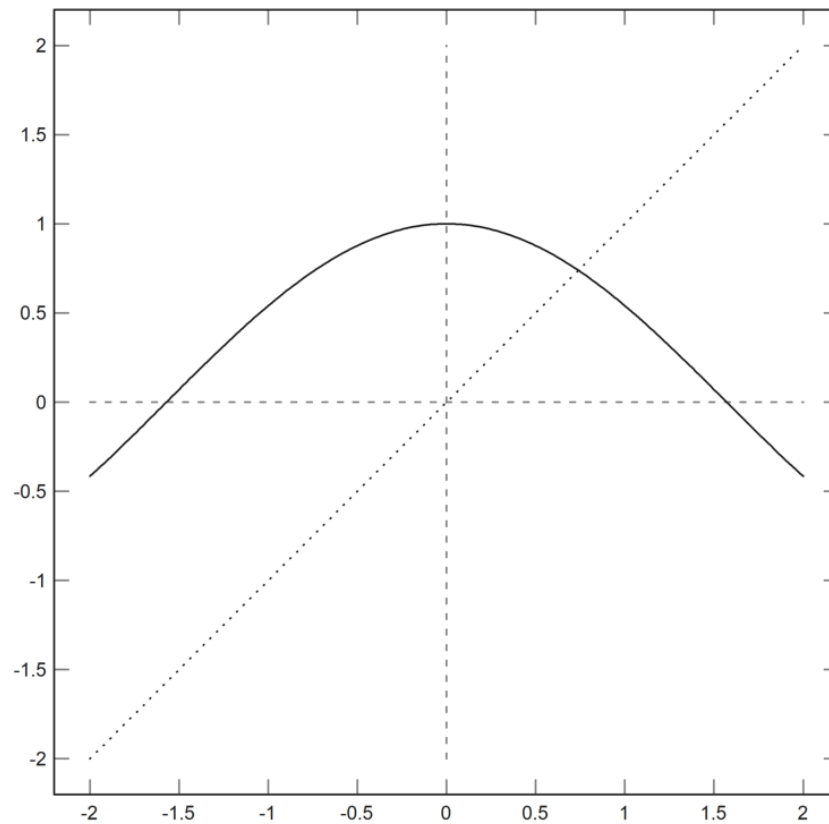


```
>columnspot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```

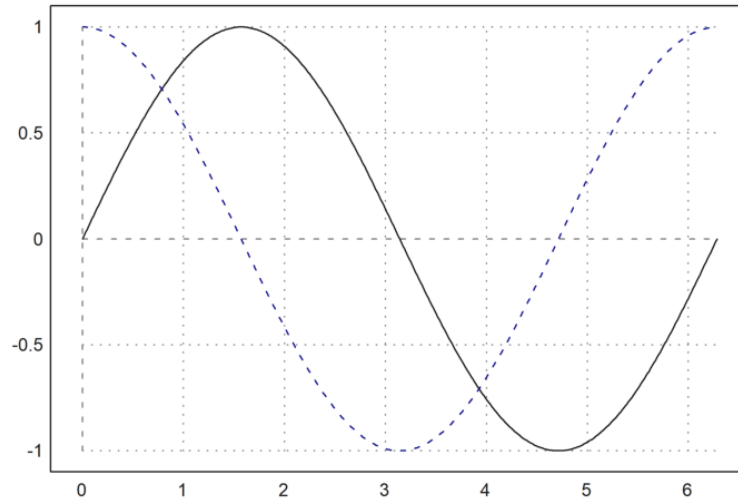


- Cara menggambar beberapa kurva pada bidang koordinat yang sama
Contoh:

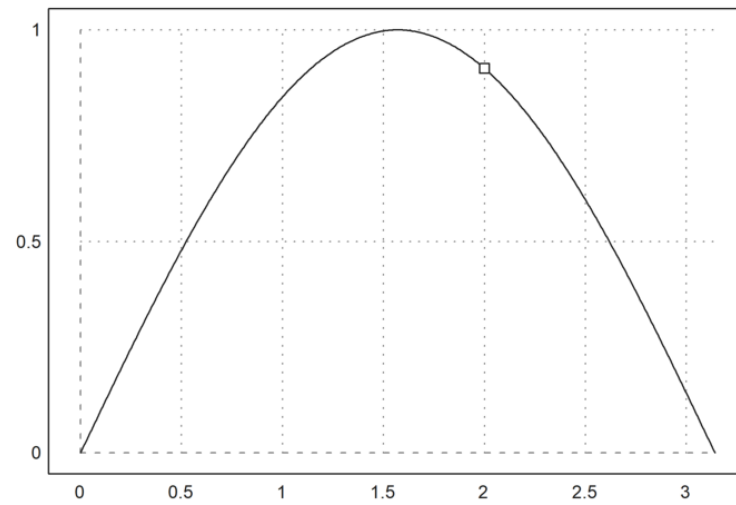
```
>aspect(); plot2d("cos(x)",r=2,grid=6); plot2d("x",style=".",>add):
```



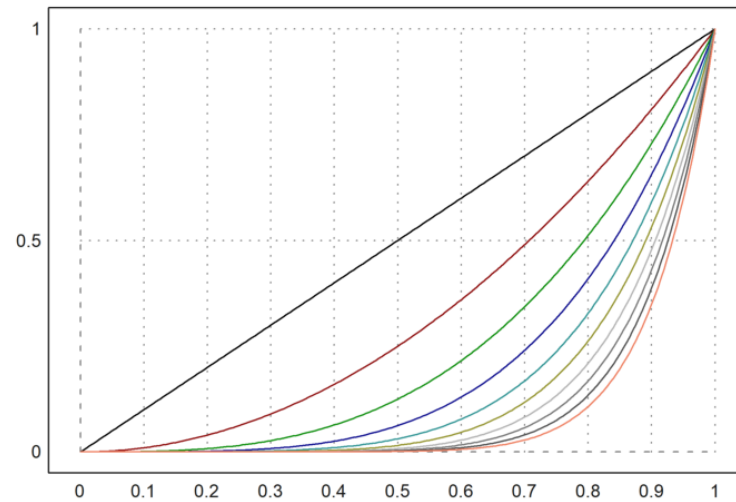
```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



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

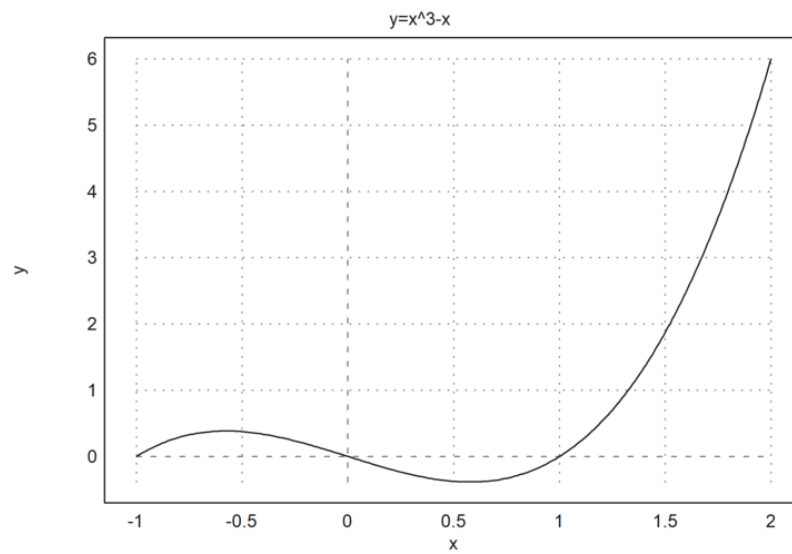


```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
```

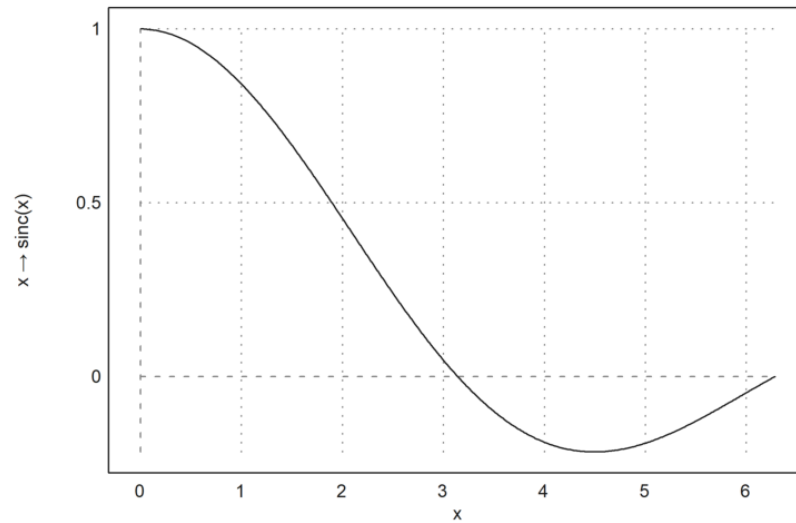


- Cara menambahkan label teks
Contoh:

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```



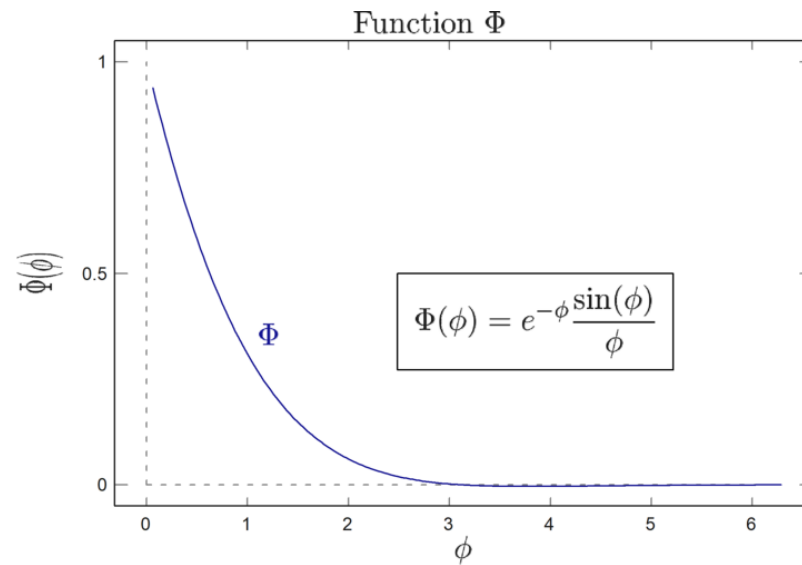
```
>plot2d("sinc(x)",0,2pi,xl="x",yl=u"x &rarr; sinc(x)",>vertical):
```



- Cara membuat latex dalam grafik

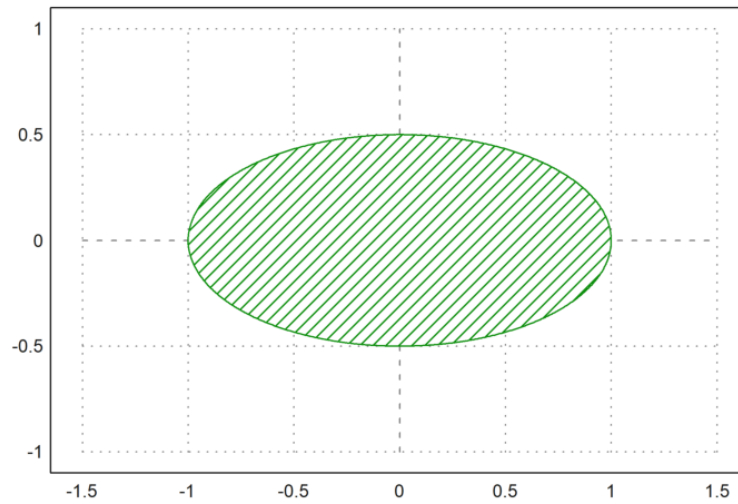
Contoh:

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

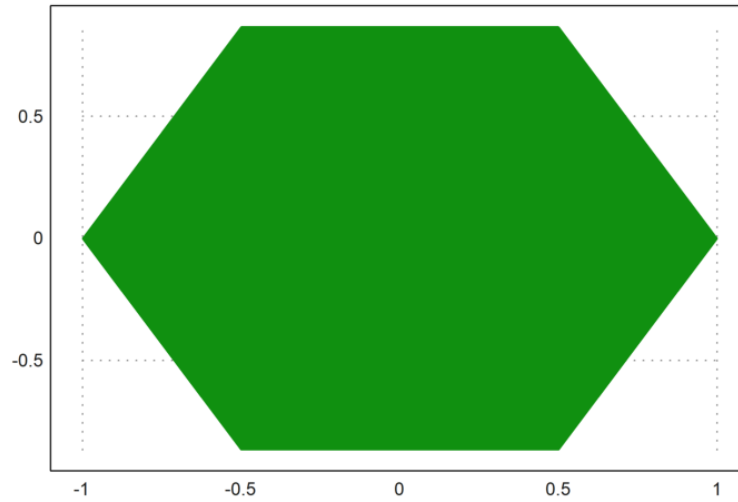


- Cara menggambar daerah yang dibatasi kurva
Contoh:

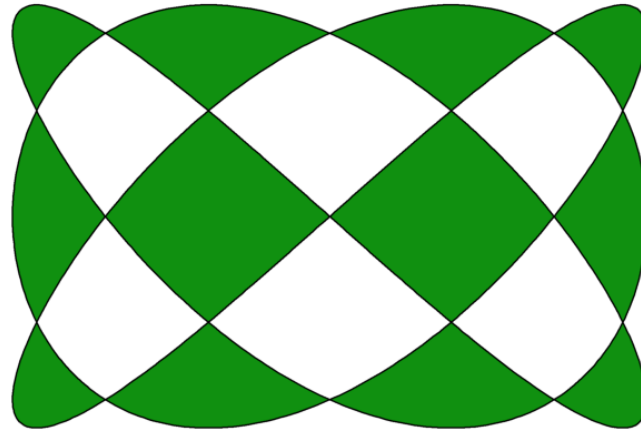
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



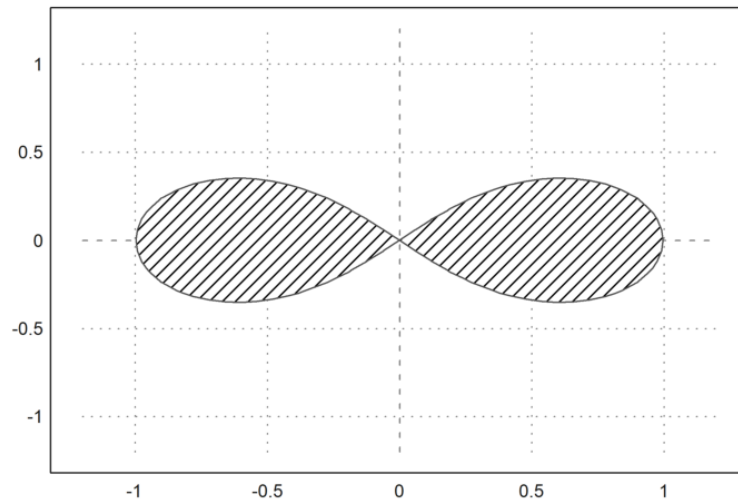
```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);  
>plot2d(x,y,<grid,<frame,>filled):
```

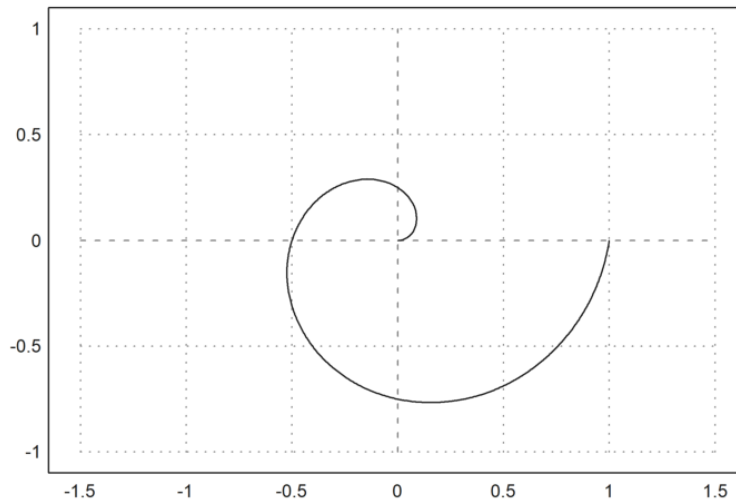


```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
```



- Cara menggambar grafik fungsi parametrik
Contoh:

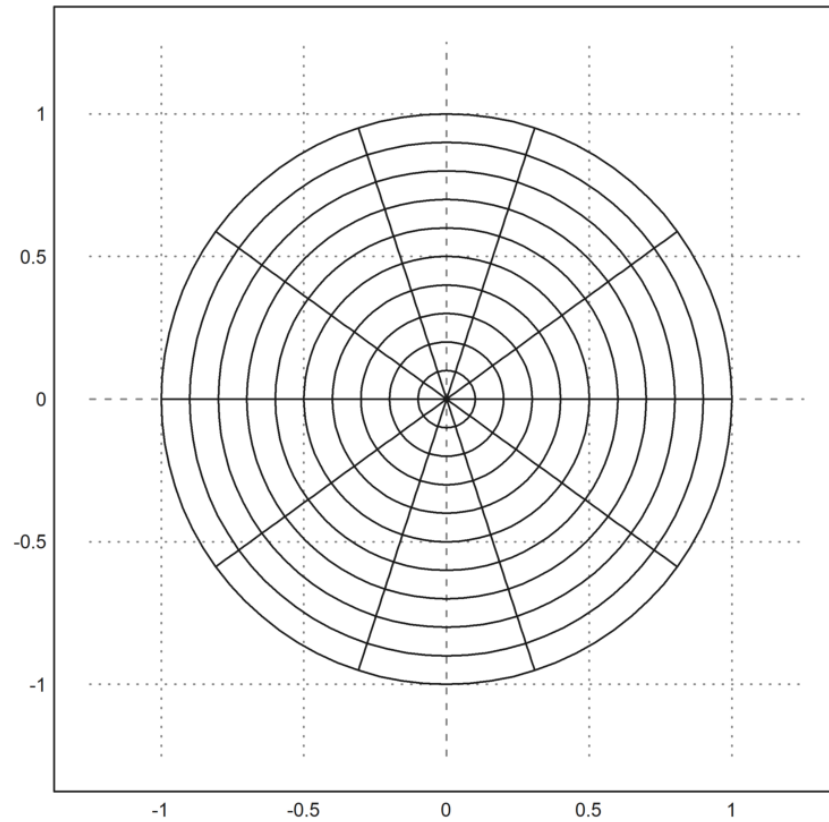
```
>plot2d("x*cos(2*pi*x)", "x*sin(2*pi*x)", xmin=0, xmax=1, r=1):
```



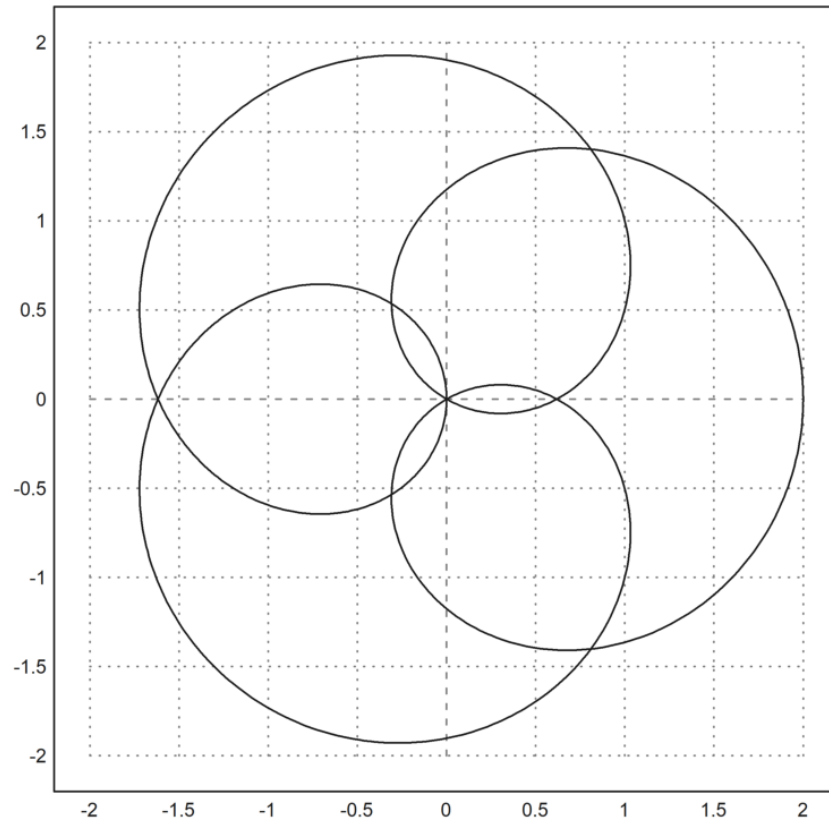
- Cara menggambar grafik bilangan kompleks

Contoh:

```
>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):
```

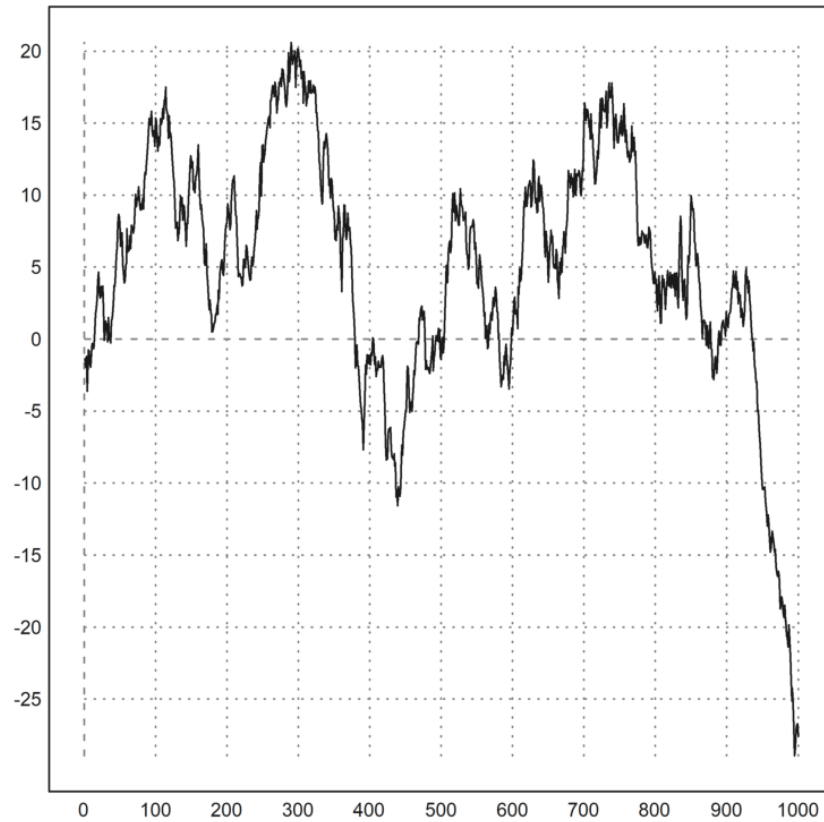


```
>t=linspace(0,2pi,1000); ...  
>plot2d(exp(I*t)+exp(4*I*t),r=2):
```

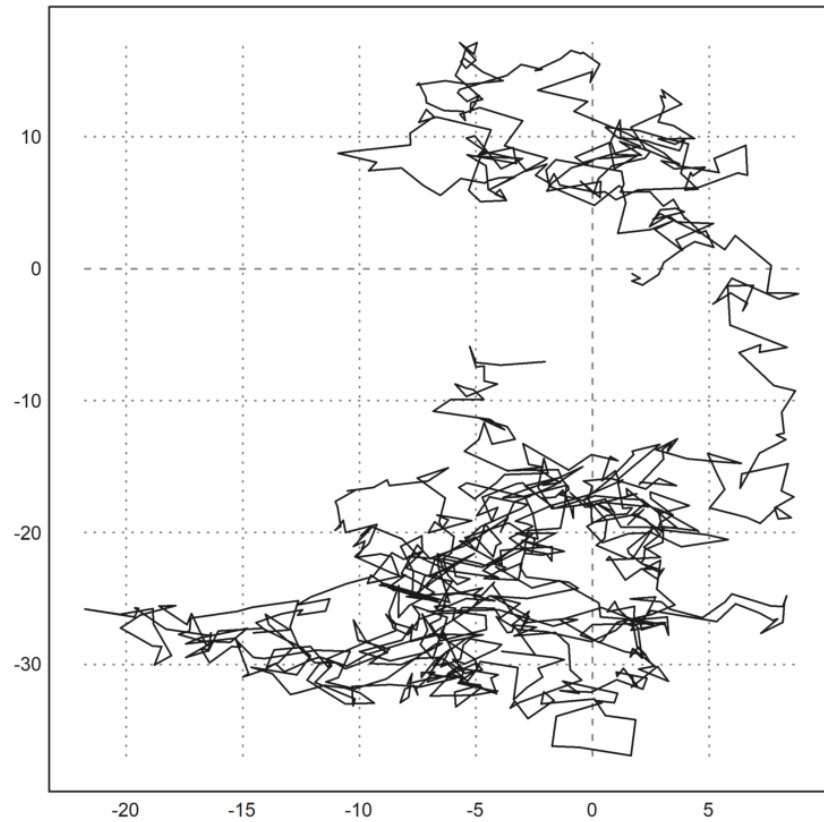


- Cara menggambar plot statistik
Contoh:

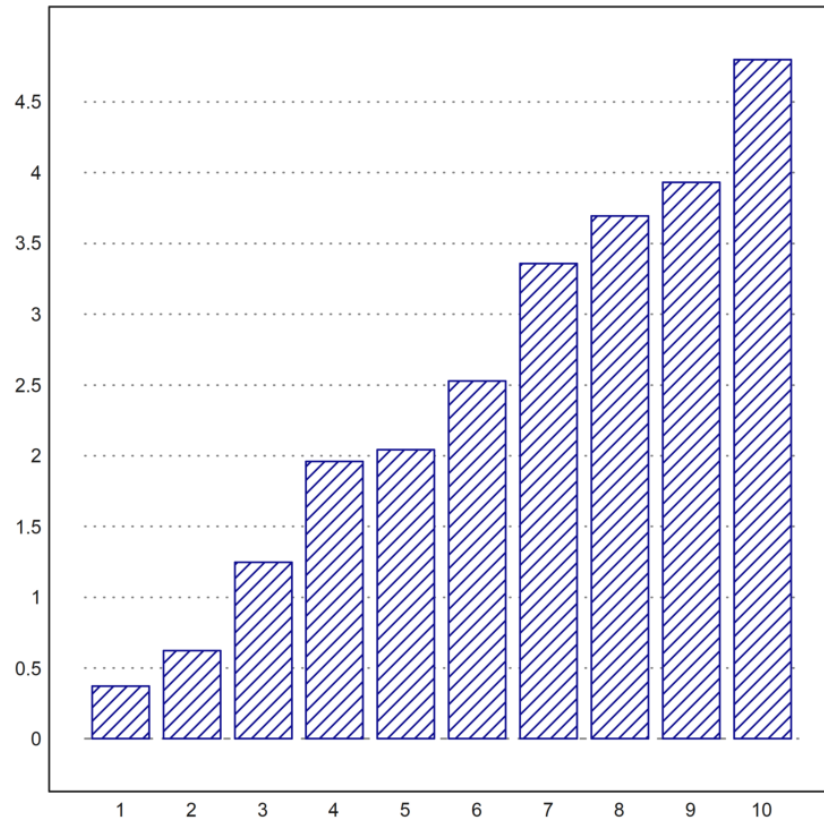
```
>plot2d(cumsum(randnormal(1,1000))):
```



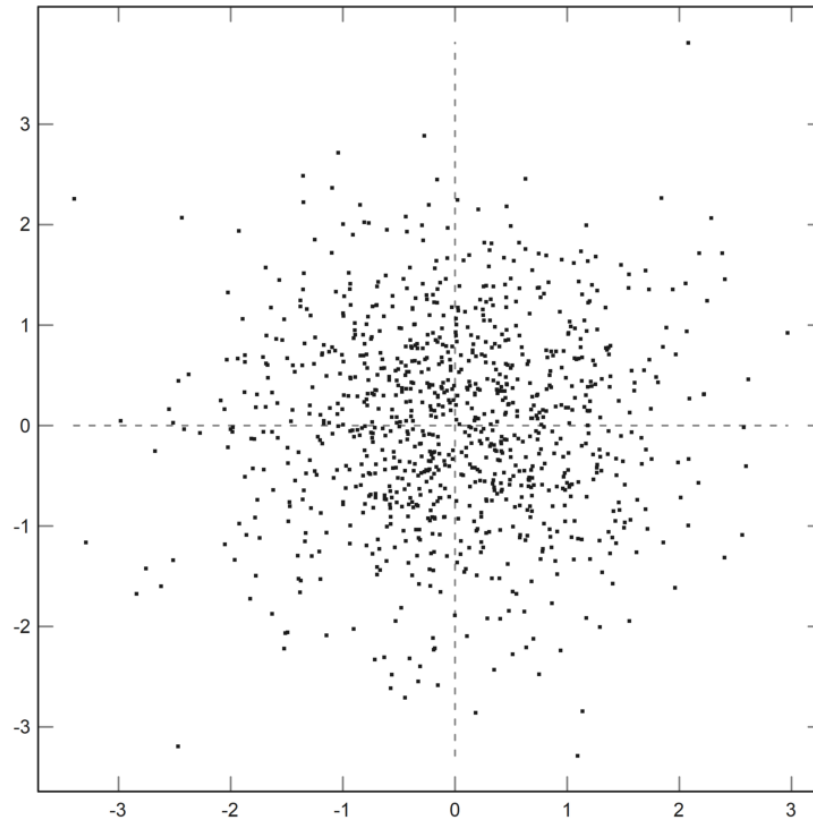
```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
```



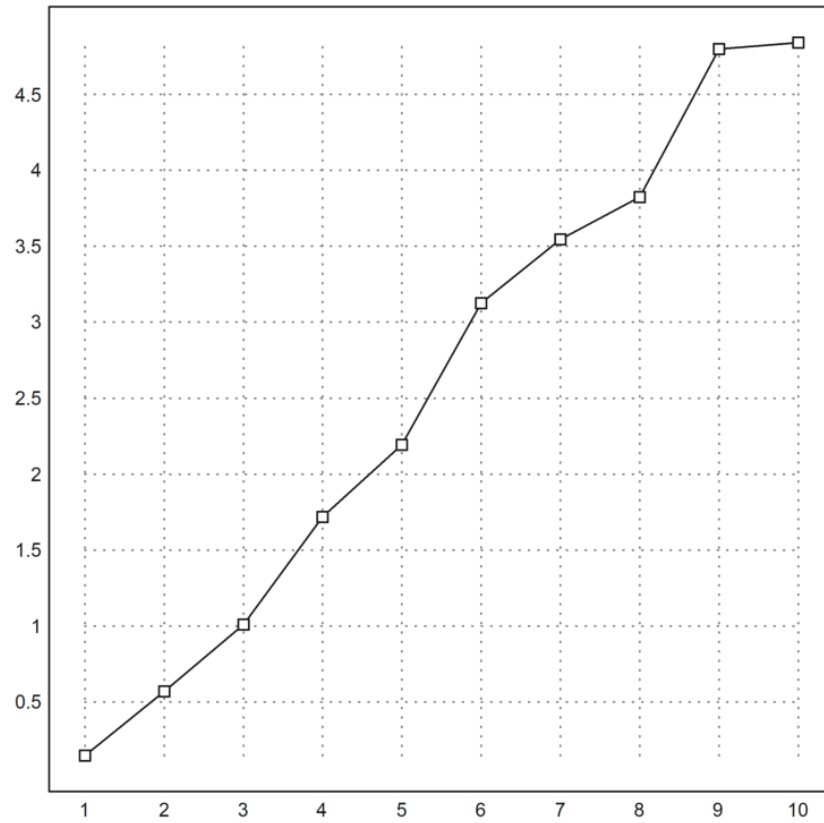

```
>columnspot(cumsum(random(10)),style="/",color=blue):
```



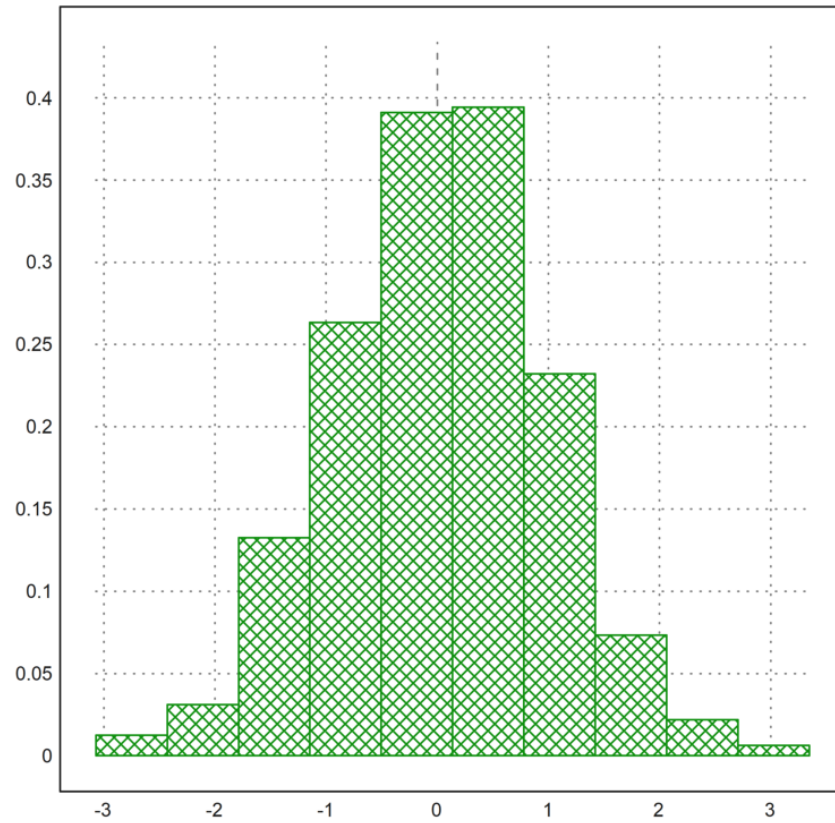
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



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

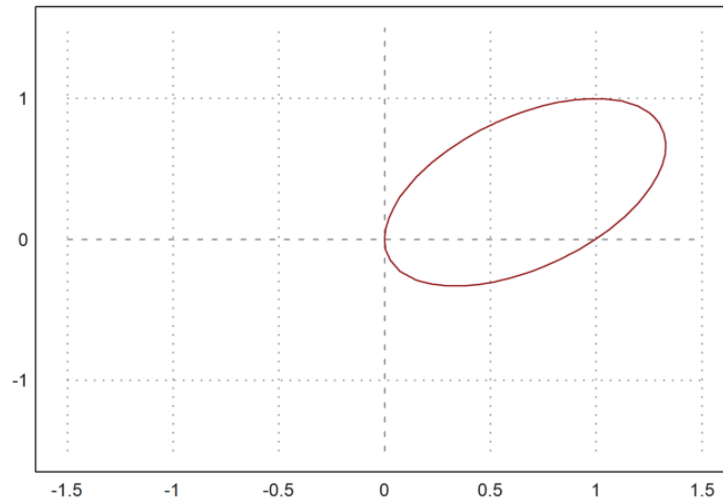


```
>plot2d(normal(1,1000),distribution=10,style="\/"):
```

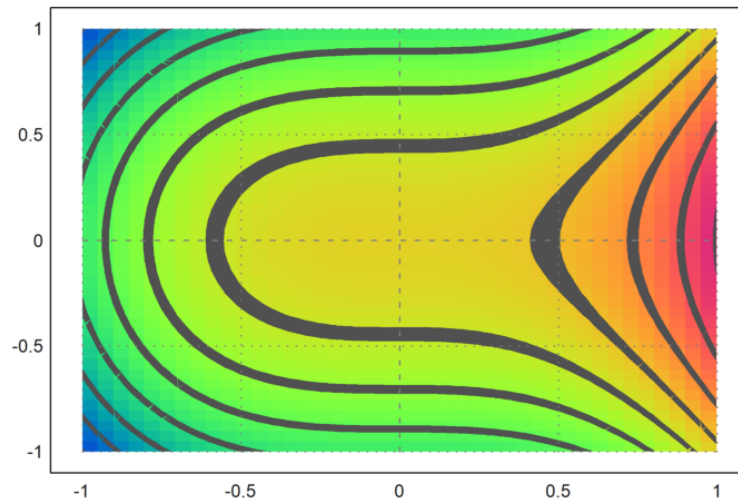


- Cara menggambar grafik fungsi implisit
Contoh:

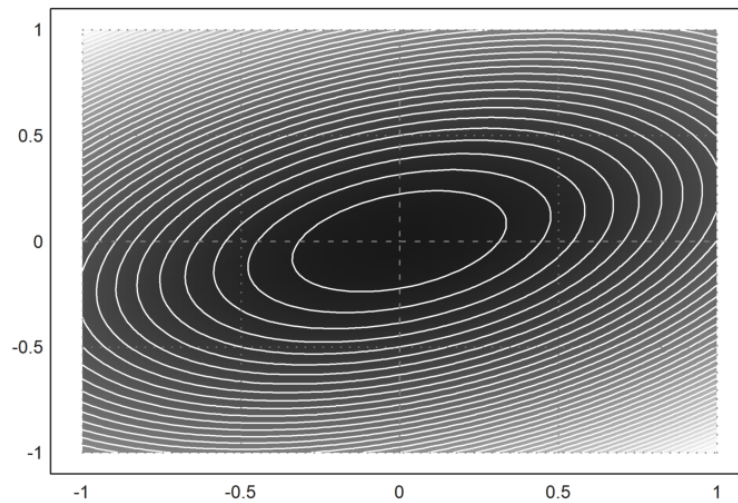
```
>aspect(1.5);  
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



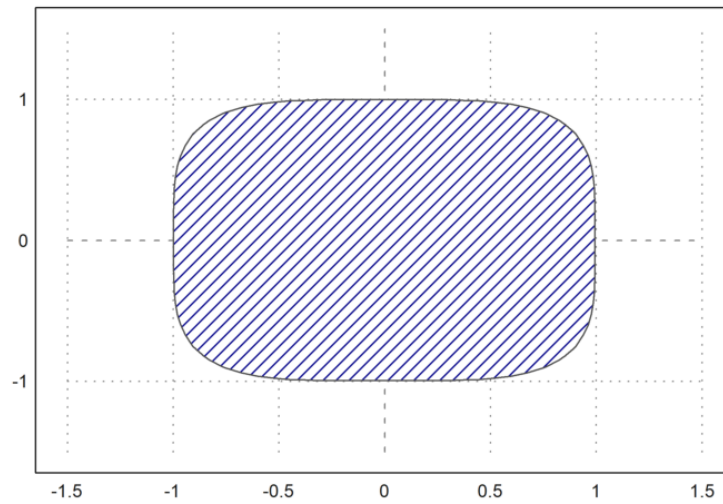
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



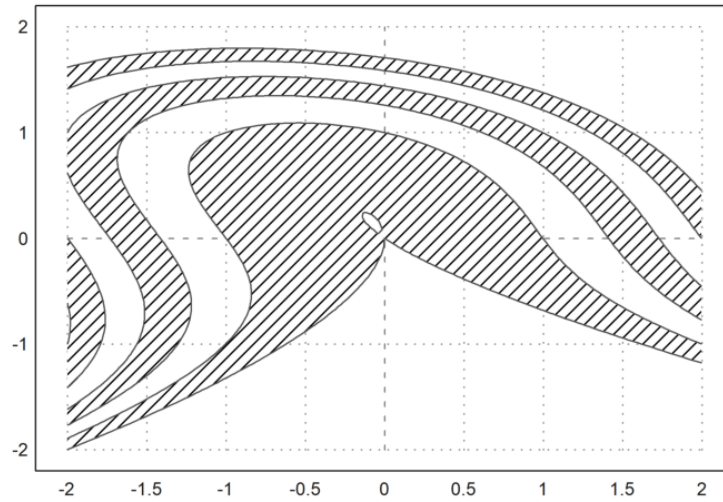
```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```



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

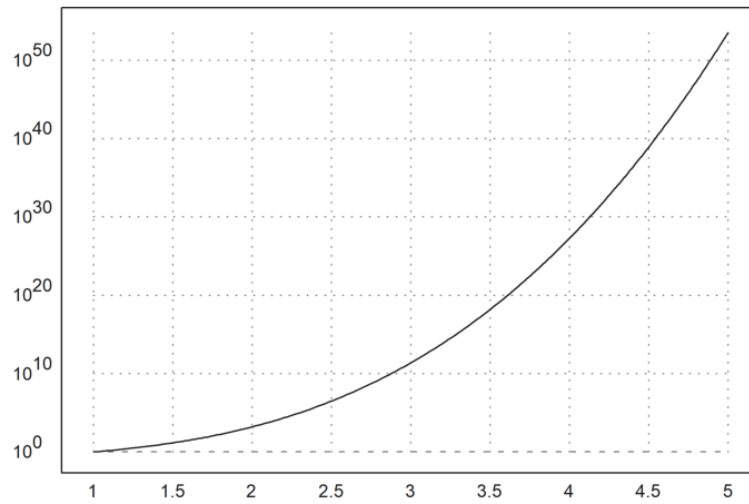


```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```

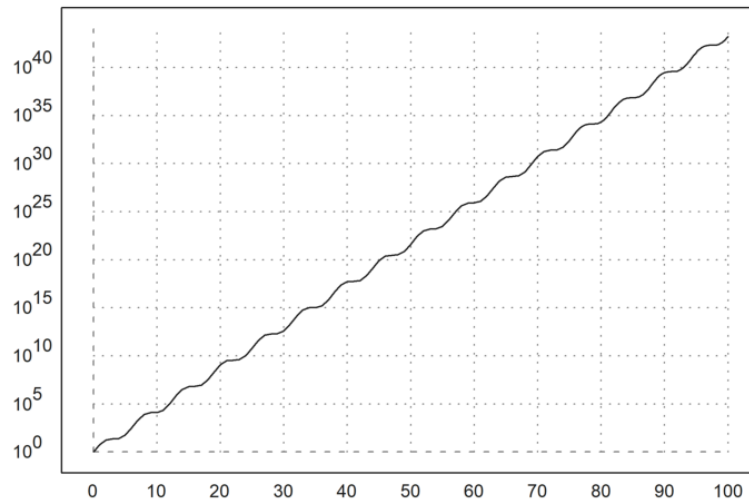



- Cara menggambar plot logaritmik
Contoh:

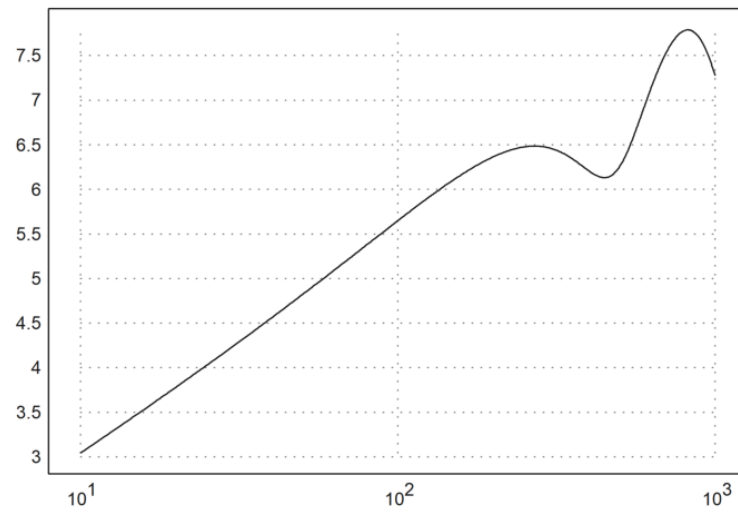
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



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



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



(BAB 5) PENGGUNAAN SOFTWARE EMT UNTUK MENGGAMBAR GRAFIK 3 DIMENSI

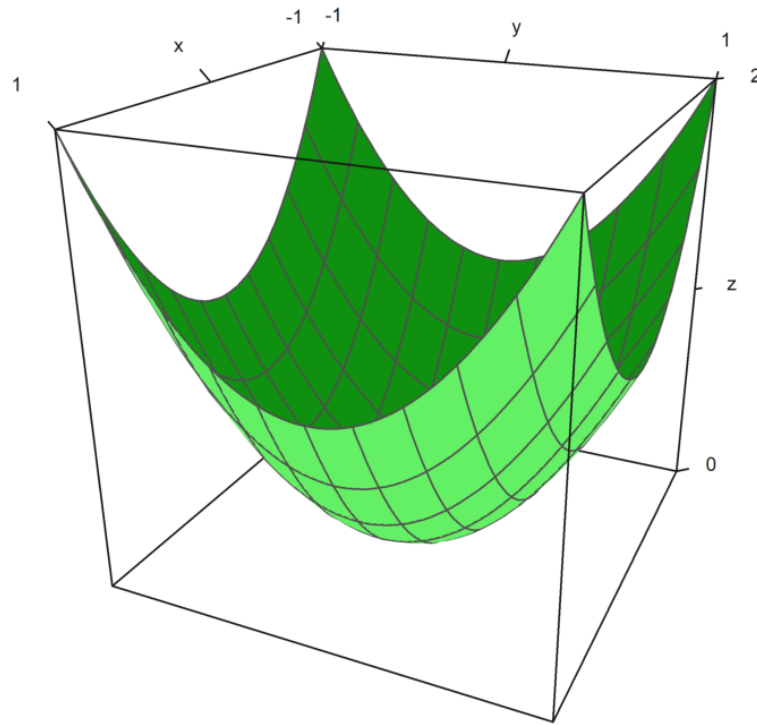
(3D)

Dalam materi ini kami diperkenalkan dengan syntax lainnya yaitu PLOT3D yakni perintah untuk menghasilkan gambar grafik terhadap BIDANG KARTESIUS 3 DIMENSI atau BIDANG-XYZ dimana gambar yang dihasilkan dari suatu permasalahan matematika, akan sangat sulit digambarkan visualisasinya dengan grafik jika SOAL MATEMATIKA tersebut dalam variabel (x,y,z) sehingga dengan bantuan PLOT 3D memudahkan saya untuk menyelesaikan persoalan tersebut dengan cepat.

Beberapa syntax dan perintah yang utama digunakan dalam PLOT 3D diantaranya adalah sebagai berikut :

- cara menggambar grafik fungsi 2 Variabel dengan ekspresi langsung
- Untuk grafik suatu fungsi, gunakan
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data.

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



- mengetahui fungsi umum untuk Plot 3D

Fungsi plot3d (x, y, z, xmin, xmax, ymin, ymax, n, a, ..

b, c, d, r, scale, fscale, frame, angle, height, zoom, distance, ..)

Rentang plot untuk fungsi dapat ditentukan dengan

- a,b: rentang x

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

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

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

Tampilan dapat diubah dengan berbagai cara.

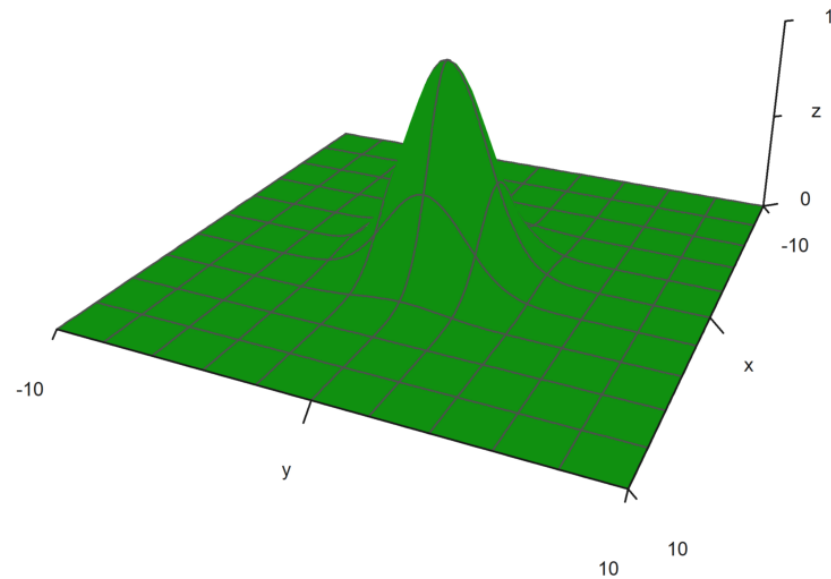
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian pandangan dalam radian.

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

```
>view
```

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

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



- cara menggambar fungsi dua variabel yang rumusnya disimpan dalam variabel ekspresi
Fungsi ini dapat memplot plot 3D dengan grafik fungsi dua variabel, permukaan berparameter, kurva ruang, awan titik, penyelesaian persamaan tiga variabel. Semua plot 3D bisa ditampilkan sebagai anaglyph.

fungsi plot3d (x, y, z, xmin, xmax, ymin, ymax, n, a

Parameter

x : ekspresi dalam x dan y

x,y,z : matriks koordinat suatu permukaan

x,y,z : ekspresi dalam x dan y untuk permukaan parametrik

x,y,z : ekspresi dalam x untuk memplot kurva ruang

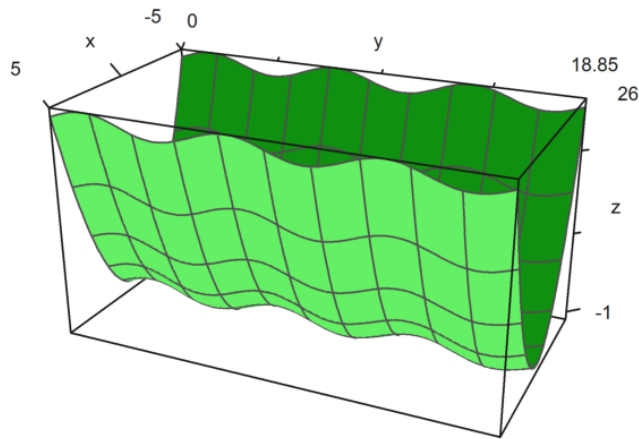
xmin,xmax,ymin,ymax :

x,y batas ekspresi

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

x^2+sin(y)

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



- cara menggambar fungsi dua variabel yang fungsinya didefinisikan sebagai fungsi numerik

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

Fungsi numerik 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

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

Untuk Gambar Grafik Fungsi

Fungsi satu baris numerik didefinisikan oleh ":=".

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

1. Buat fungsi numerik yang akan digunakan untuk memvisualisasikan data.
function $f(x,y):=ax+by$
dimana a dan b adalah konstanta
2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi numerik.
plot3d("f"):

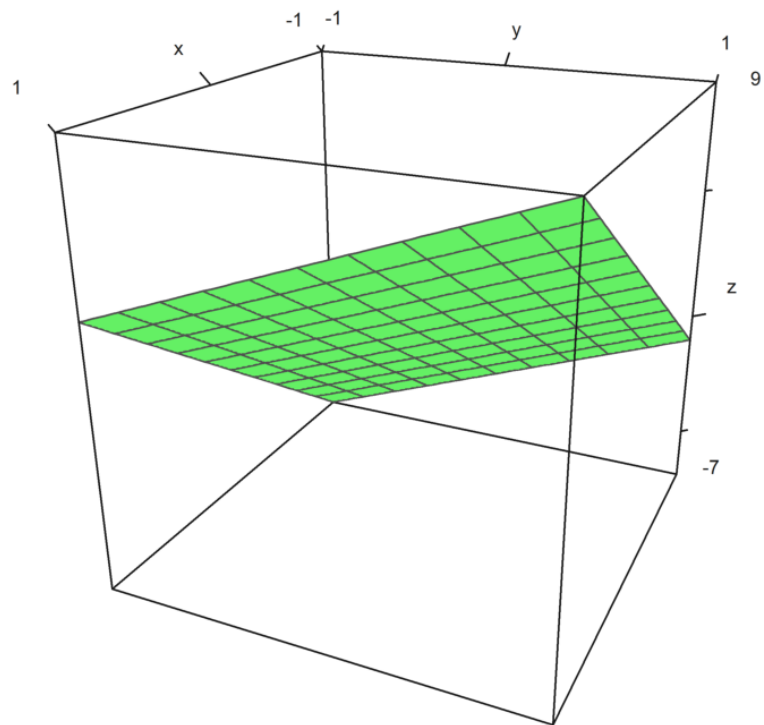
Contoh

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

1. Fungsi Linear Dua Variabel

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

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



- cara menggambar grafik fungsi dua variabel yang fungsinya didefinisikan sebagai fungsi simbolik
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 numerik.

`plot3d("g")`:

3. Menentukan rentang variabel

misal

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

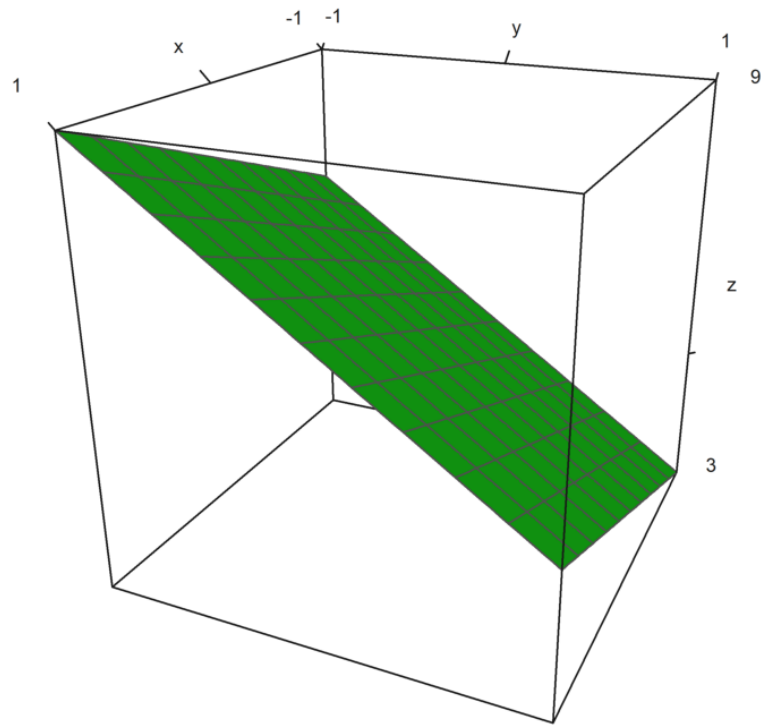
dengan batasan x dari -10 hingga 10 dan batasan y dari -5 hingga 5

Contoh

1. Fungsi Linear Dua Variabel

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

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



- Cara 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 $f_x(x,y)$, $f_y(x,y)$, $f_z(x,y)$.

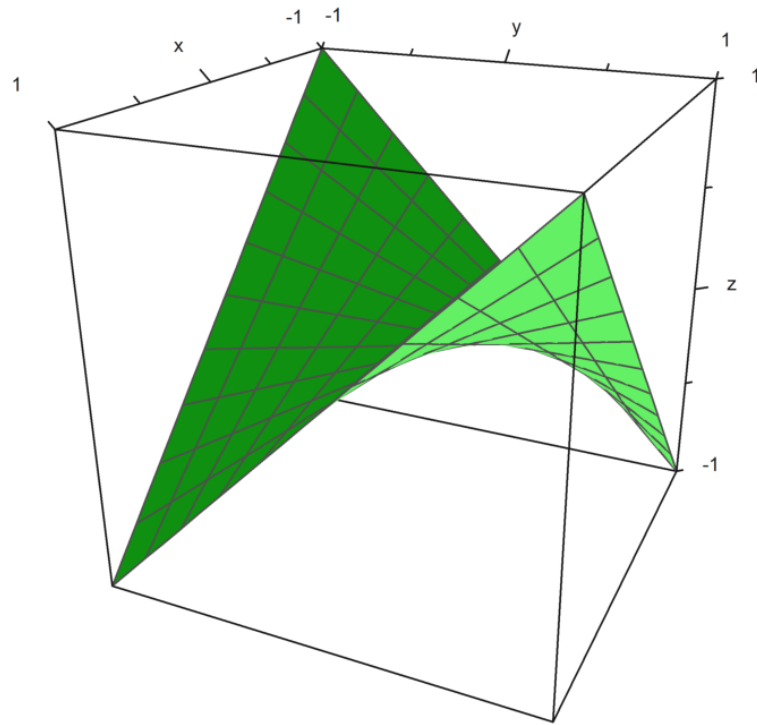
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

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.

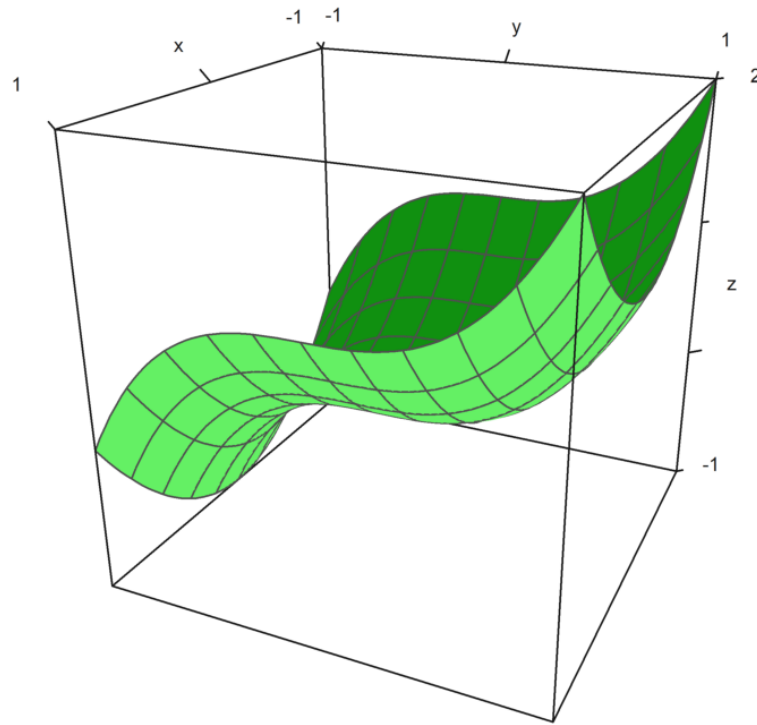
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



- cara menggambar 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.

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



- Cara menggambar grafik fungsi parametrik 3D

Fungsi parametrik merupakan jenis fungsi matematika yang menggambarkan hubungan antara dua atau lebih variabel, dimana masing-masing koordinat (x, y, z, \dots) 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.

```
>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):
```

Closing bracket missing in function call!

Error in:

```
>n=90,grid=[20,50],wirecolor=black,zoom=5): ...  
^
```

- Cara menggambar fungsi 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.

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

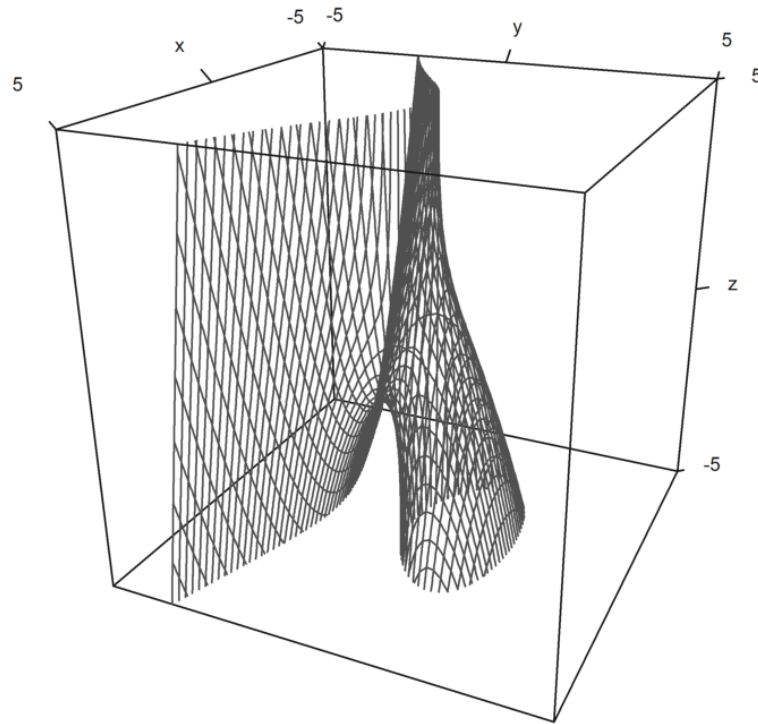
(1 persamaan dan 3 variabel), terdiri dari 2 variabel bebas dan 1 terikat

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):
```



- cara menggambar fungsi implisit menggunakan povray

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

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

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

Dalam materi ini, sama seperti materi PLOT2D dimana kita memasukkan syntax sesuai plot yang kita inginkan berdasarkan **PERSOALAN** atau **PERMASALAHAN MATEMATIKANYA**. Disini saya telah menggunakan beberapa syntax untuk mengaplikasikannya dengan **SOAL SOAL LATIHAN** pada buku **ALJABAR** dan membuat baris perintah baru yang sesuai dengan **PLOT3D**.

(BAB 6) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI GEOMETRI

Dalam materi ini, sama halnya dengan PLOT2D dan PLOT3D kami mempelajari cara menggambar dalam hal ini adalah pada materi Geometri yang telah mengaplikasikan banyak permasalahan matematika kedalam visualisasi maupun gambar. Geometri juga memiliki FITUR khusus pada EMT yang dapat dipanggil, yakni sebagai berikut :

- cara memanggil fungsi geometri agar dapat menjalankan perintah

```
> load geometry
```

```
Numerical and symbolic geometry.
```

- mengetahui fungsi-fungsi Geometri pada EMT
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

$\text{quad}(A,B)$: kuadrat jarak AB

$\text{spread}(a,b,c)$: Spread segitiga dengan panjang sisi-sisi a,b,c , yakni

$\sin(\alpha)^2$ dengan α sudut yang menghadap sisi a .

$\text{crosslaw}(a,b,c,sa)$: persamaan 3 quads dan 1 spread pada segitiga

dengan panjang sisi a, b, c .

$\text{triplespread}(sa,sb,sc)$: persamaan 3 spread sa,sb,sc yang membentuk

suatu segitiga

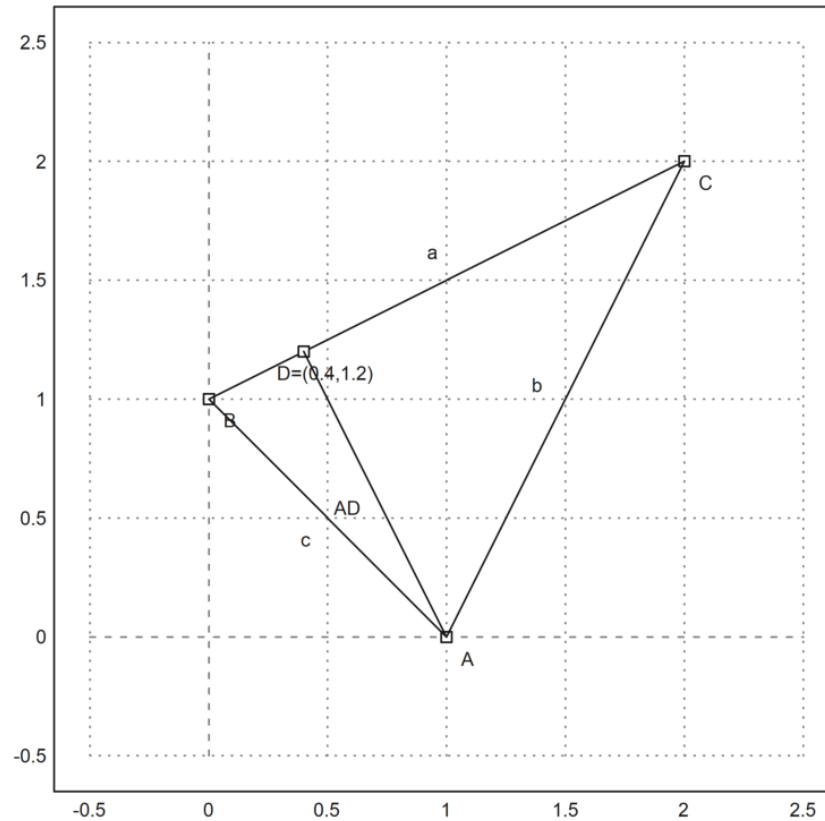
$\text{doublespread}(sa)$: Spread sudut rangkap Spread 2ϕ , dengan

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

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
>lineThrough(B,C) // garis yang melalui B dan C
```

[-1, 2, 2]

```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()
```



- cara menggambar geometri simbolik Kita dapat menghitung geometri tepat dan simbolis menggunakan Maxima.

Geometri file.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, sekarang kita dapat menggunakan perhitungan simbolik.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
>c &= lineThrough(B,C) // c=BC
```

$[-1, 2, 2]$

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$2y - x = 2$$

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

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

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

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

$[2, 1, 2]$

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

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

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

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

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

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

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

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

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

$$\frac{5}{3\sqrt{2}}$$

$$1.178511301977579$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi s
```

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

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

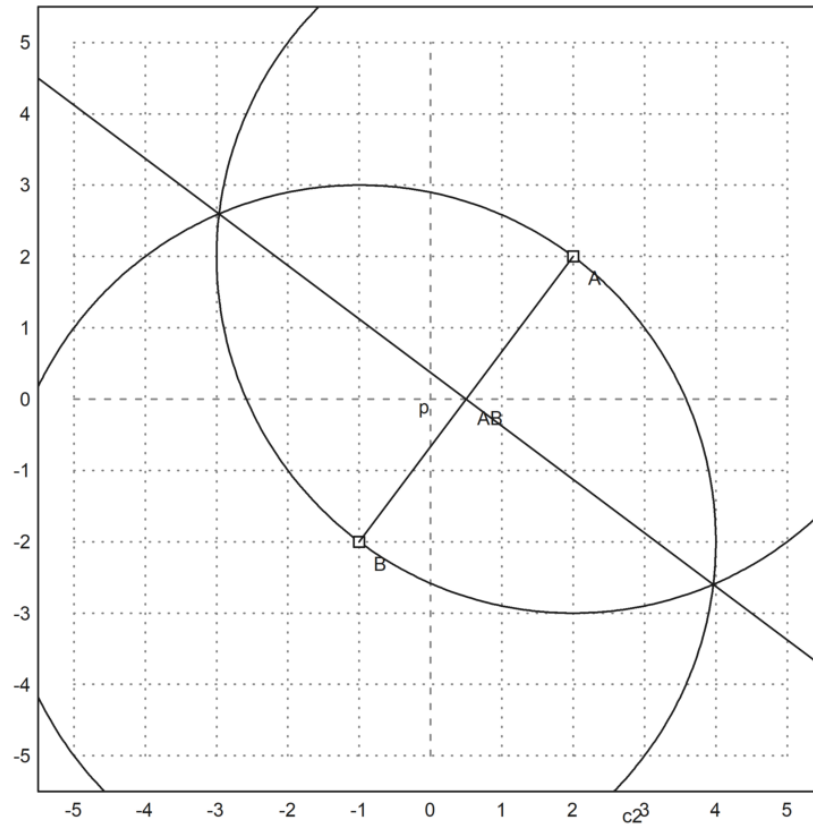
```
[0.86038, 0.86038]
```

- cara menggambar garis sumbu

Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis 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):
```

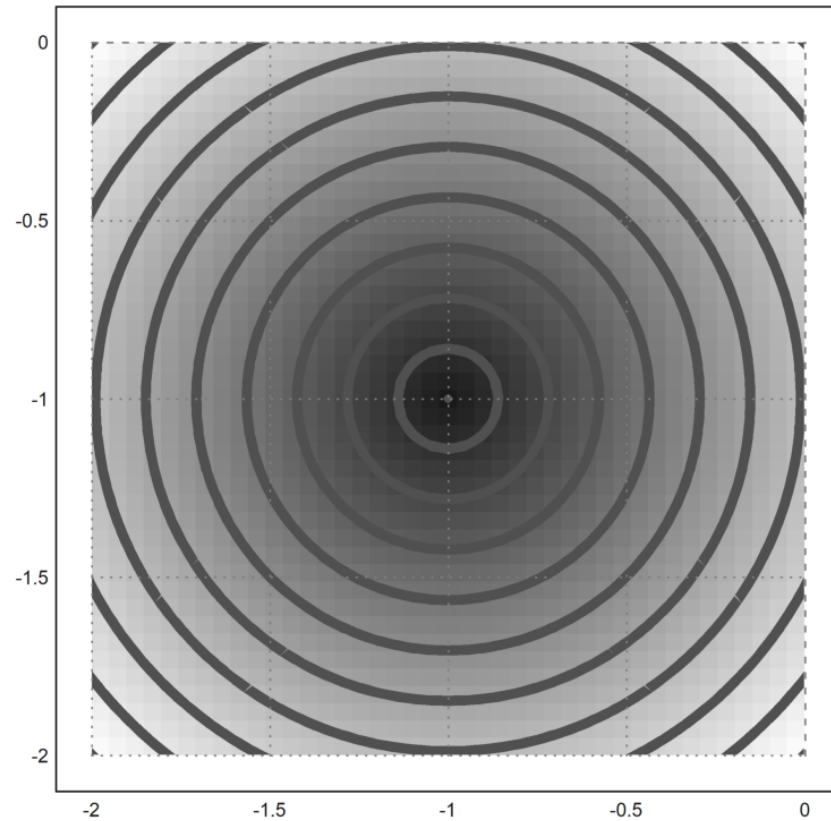


- cara menghitung jarak minimal pada bidang

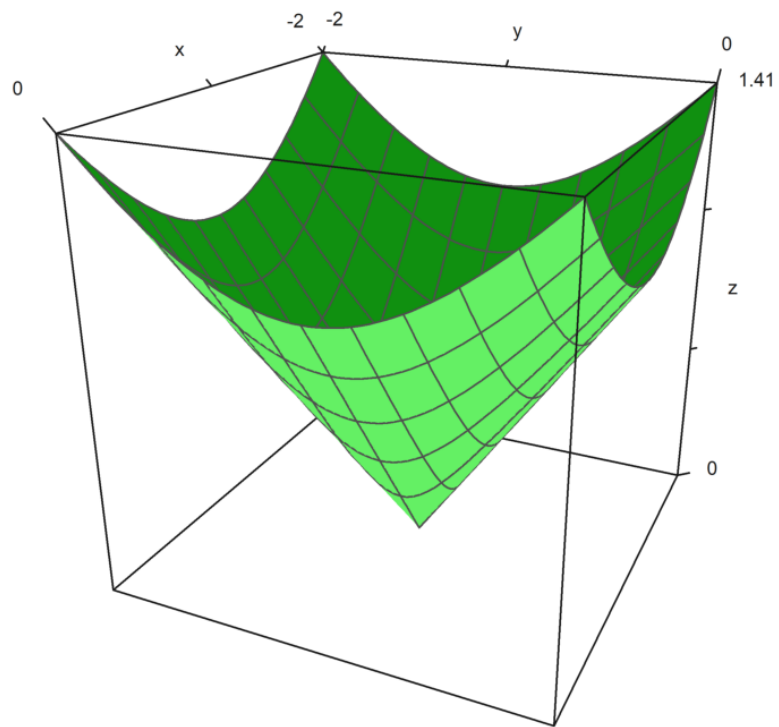
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M , memiliki garis level yang agak sederhana: lingkaran berpusat di A .


```
>&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");
```

If you see ellipses, please set your window square



```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```



- cara menghitung Geometri Bumi
Fungsi-fungsi tersebut terdapat dalam file "spherical.e"

```
>load "spherical.e";  
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY, sposprint untuk cetak posisi bo
```

S 7°46.467' E 110°23.050'

(b) Hal-hal yang dilakukan dalam mempelajari materi

- mencoba perintah dengan fungsi geometri
- mencari sumber informasi lain
- mencoba menyelesaikan soal geometri
- membuktikan apakah fungsi benar

(c) Kendala-kendala yang dihadapi dan usaha-usaha yang dilakukan

- fungsi tidak lengkap. usaha yang dilakukan menggabungkan rumus-rumus yang menghasilkan fungsi baru
- terjadi error saat menjalankan perintah karena tidak memanggil fungsi geometri, usaha yang dilakukan memanggil fungsi geometri sebelum menjalankan fungsi yang lain

(BAB 7) PENGGU-

NAAN SOFTWARE EMT UNTUK APLIKASI KALKULUS

1. Untuk 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*x^2+exp(sin(x)) // fungsi numerik  
>f(0), f(1), f(pi)
```

```
1  
4.31977682472  
20.7392088022
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format `:=`, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

2. Menentukan fungsi 1 variabel

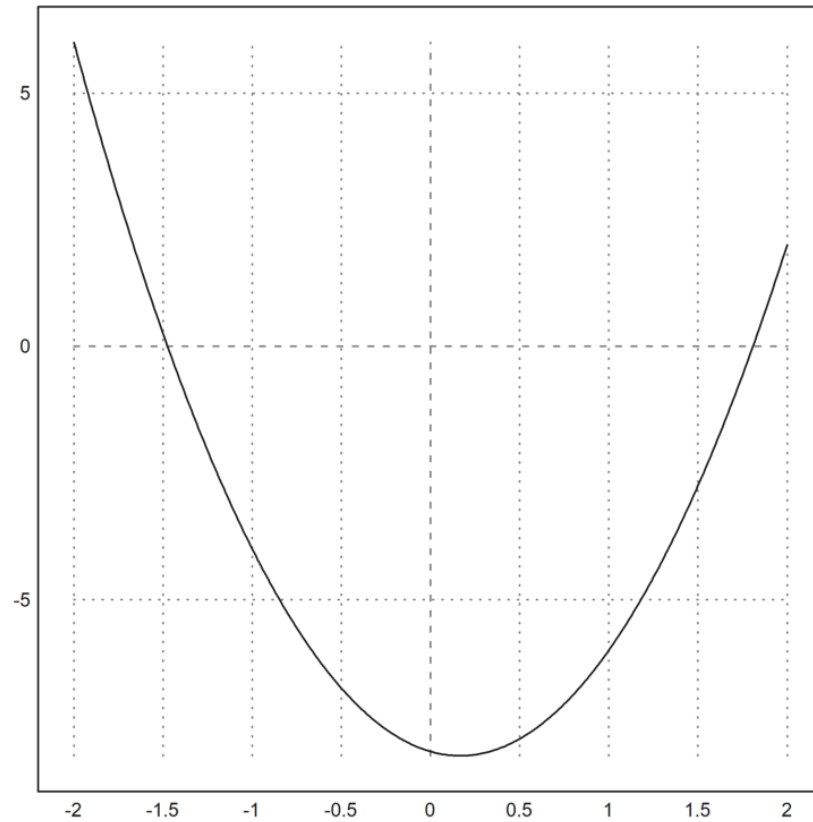
```
>function p(x) := 3*x^2-x-8  
>p(0), p(4), p(6)
```

```
-8  
36  
94
```

```
>pmap(0:2)
```

```
[-8, -6, 2]
```

```
>plot2d("p(x)":
```



3. Menentukan fungsi 2 variabel

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

```
return sqrt(x^2+y^2)
endfunction
```

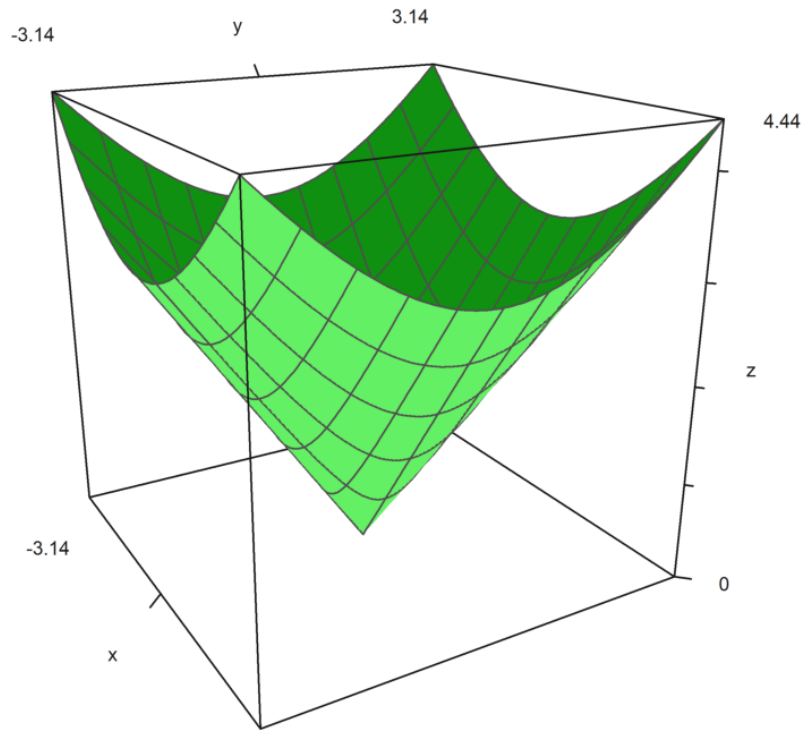
```
>f(0,5), f(4,8), f(-1,4)
```

```
5
8.94427191
4.12310562562
```

```
>fmap(-3:0,1:4)
```

```
[3.16228,  2.82843,  3.16228,  4]
```

```
>aspect=1.5; plot3d("f(x,y)",a=-80,b=80,c=-80,d=80,angle=60°,height=20°,r=pi,n=100):
```

4. 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 ($-\infty$, \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).
Contoh:

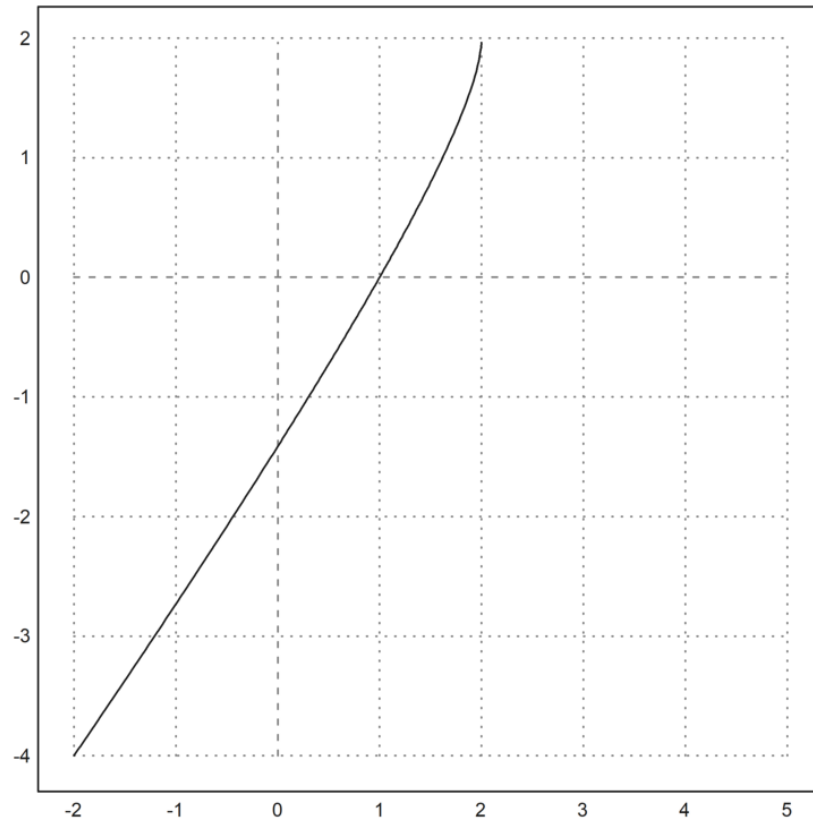
```
>$showev('limit(1/(2*x-1),x,0))
```

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

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

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

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



5. Menentukan Turunan fungsi
Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>function f(x) := (x^5)
>$showev('limit((((x+h)^5)-x^5)/h,h,0)) // Turunan x^5
```

Maxima said:

limit: second argument must be a variable, not a constant; found:

[2,1,2]

#0: showev(f='limit([(x+2)^5-x^5)/2,(x+1)^5-x^5,((x+2)^5-x^5)/2],[2,1,2],0))

-- an error. To debug this try: debugmode(true);

Error in:

```
$showev('limit((((x+h)^5)-x^5)/h,h,0)) // Turunan x^5 ...
^
```

6. Digunakan untuk menghitung integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

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

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

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

Answering "Is n 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))
```

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

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

7. Menghitung volume perputaran kurva

$$m(x) = x^2 + 2$$

dari $x=0$ sampai $x=1$. Diputar terhadap sumbu- x .

Jawab:

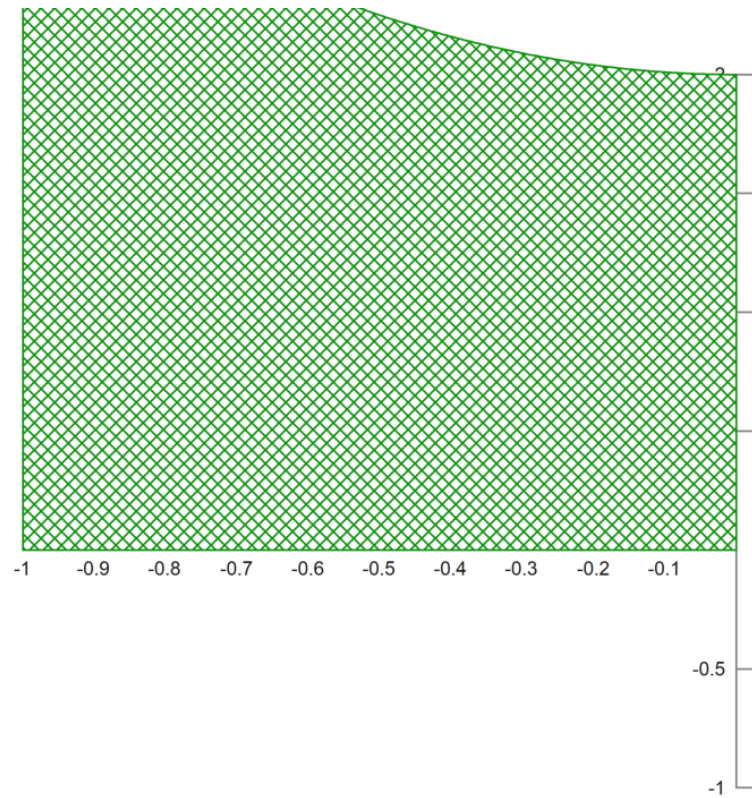
```
>function m(x) &= x^2+2; $m(x)
```

$$x^2 + 2$$

```
>$showev('integrate(m(x),x,-1,1))
```

$$\int_{-1}^1 x^2 + 2 \, dx = \frac{14}{3}$$

```
>plot2d("m(x)",-1,0,-1,2,grid=7,>filled, style="/\"):
```



8. Menentukan barisan dan deret

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 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
```

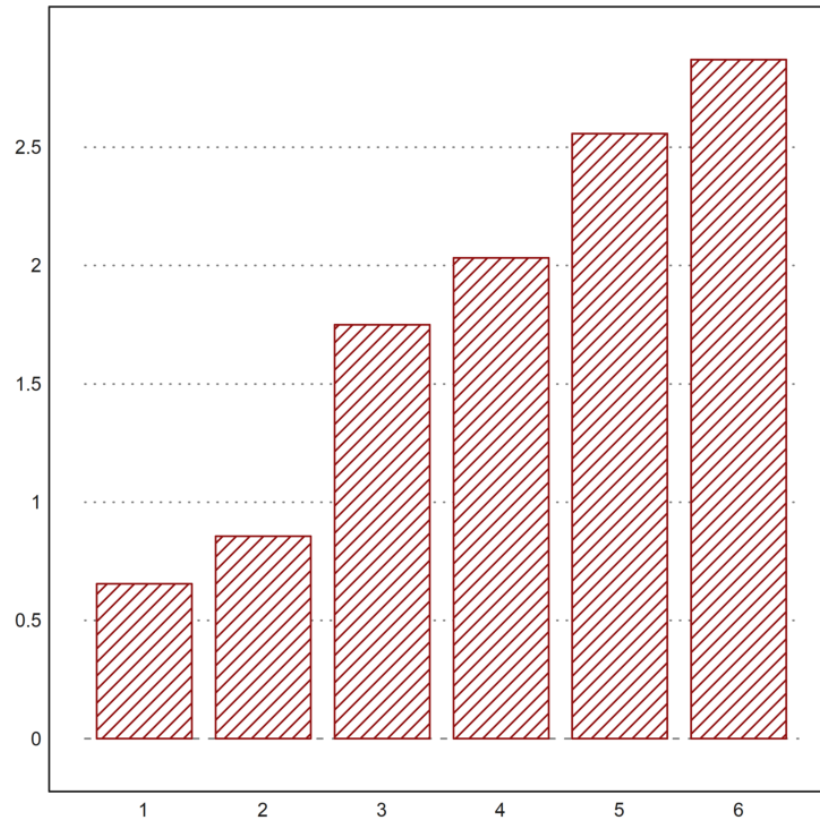
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```


(BAB 8) PENGGUNAAN SOFTWARE EMT UNTUK APLIKASI STATISTIKA

1. Menggambar grafik statistika(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, dengan contoh sebagai berikut :

```
>columnplot(cumsum(random(6)),style="/",color=red):
```



2. 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.

3. 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.

4. Mencari simpangan baku

Standar Deviasi atau simpangan baku adalah akar dari ragam/varians. Untuk menentukan nilai standar deviasi, caranya:

$$\sigma = \sqrt{\sigma^2}$$

atau

$$S = \sqrt{S^2}$$

5. Mencari jangkauan

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.

contoh :

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)
```

6. 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)
```

Contoh:

Mentukan 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]
```

(b) Hal-hal yang saya lakukan dalam mempelajari materi tersebut

- mencari sumber materi yang lebih lengkap
- mencoba perintah-perintah baru mengenai materi Statistika
- menyelesaikan soal yang ada

(c) kendala-kendala yang saya hadapi dan usaha-usaha yang saya lakukan

- terjadi error saat menjalankan perintah, usaha yang dilakukan mengecek perintah dan melengkapi kesalahan

(BAB 9) PENGOLAHAN DOKUMEN MENGGUNAKAN LATEX

(a) HAL-HAL YANG SAYA PELAJARI

Dalam materi ini, saya menjadi tahu mengenai beberapa cara untuk membuat DOKUMEN TERSTRUKTUR dengan menggunakan LATEX, dimana file EMT yang kita buat dapat diubah menjadi latex kemudian dengan memanfaatkan SOFTWARE OVERLEAF kita dapat mengelola dokumen LATEX yang sudah kita buat di EMT agar lebih rapi dan terstruktur.

- Cara mengolah dokumen menggunakan overleaf

contohnya : menyatukan beberapa file menjadi satu dokumen seperti yang telah dilakukan pada tugas sebelumnya

- Cara membuat repository pada akun github

contohnya : saya sudah memiliki akun github yang saya gunakan untuk memposting atau membagikan hasil pekerjaan APLIKASI KOMPUTER 9 materi saya, pada tugas Overleaf dan GITHUB minggu lalu.

(b) HAL-HAL YANG SAYA LAKUKAN DALAM MEMPELAJARI MATERI

Saya dapat mempelajari cara membuat akun pribadi di Overleaf dan github, Mencari informasi mengenai cara menggunakan overleaf dan github, Bertanya kepada teman yang lebih paham, serta saya menelusuri website serta tutorial khusus untuk mengelola FILE LATEX tersebut.

(c) KENDALA YANG SAYA HADAPI

Dikarenakan minimnya pengetahuan saya mengenai Overleaf, saya sering kesulitan mengatasi masalah LATEX yang tidak terdeteksi, dimana hal tersebut dikarenakan kesalahan penulisan SYNTAX yang tidak sesuai sehingga hasil perintah tidak dapat dijalankan.

Selain itu sering terjadi EROR yang memunculkan tulisan "LIMIT PENGGUNAAN OVERLEAF TELAH MENACAPAI BATAS"

Oleh karena itu, usaha yang saya lakukan adalah memperbaiki perintah maupun syntax yang salah atau tidak sesuai, dengan demikian program OVERLEAF tidak eror lagi.